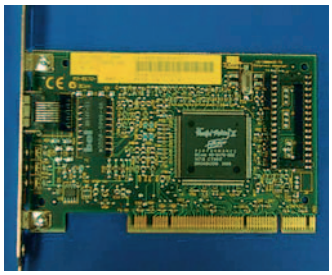
Lebih Dekat dengan Ethernet



Gunung Sarjono

(Bagian 1 dari 2 Artikel)

Semua atau hampir semua jaringan mungkin menggunakan Ethernet. Mengapa hal itu bisa terjadi? dan bagaimanakah sebenarnya Ethernet itu? Akan kita lihat pada artikel ini.

Pada artikel sebelumnya, kita telah melihat dua teknologi jaringan, FDDI dan Token Ring. Keduanya memang memiliki bagian dalam pangsa pasar LAN, tetapi bagian terbesar dipegang oleh Ethernet. Mengapa hal itu bisa terjadi? Pertanyaan itulah yang akan kita jawab pada artikel ini. Kita akan lihat akses media, topologi, dan pengalamatan Ethernet serta perbandingan komunikasi *half* dan *full duplex*.

Bagaimana Sejarah Ethernet?

Ethernet awalnya dikembangkan pada awal 1970-an oleh Xerox, dan dikembangkan lebih lanjut oleh Digital Equipment Corporation, Intel, dan Xerox (DIX) pada tahun 1980 dengan mengeluarkan Ethernet Versi 1. Pada tahun 1982, standar tersebut diperbarui dengan dikeluarkannya Ethernet Versi 2.

Pada tahun 1983, Novell membuat standar Ethernet sendiri sebelum dikeluarkannya standar IEEE 802.3. Pada tahun 1983, standar 802.3 selesai dibuat dan menjadi standar untuk koneksi Ethernet dengan *thick coax* dan *thin coax*. Pada tahun 1990, standar tersebut diperbarui dengan menambahkan Ethernet dengan *twisted pair* 10Base-T. Standar IEEE 802.3 sekarang ini meliputi *thick coax, thin coax, twisted pair* dan *fiber,* dengan kecepatan 10 Mb/s, 100 Mb/s, dan 1000 Mb/s.

Bagaimana Akses Media Ethernet?

Ethernet merupakan teknologi jaringan yang *contention-based*. Dari artikel sebelumnya, telah kita ketahui bahwa pada sistem yang contention-based, semua sistem berbagi pakai, atau berjuang mendapatkan akses untuk mengirimkan data. Tidak seperti Token Ring dan FDDI yang *deterministic*, pada lingkungan Ethernet semua sistem mendengarkan media, menunggu kesempatan untuk mendapatkan akses ke media tersebut. Hanya satu sistem yang dapat mengirimkan data pada satu waktu atau akan terjadi *collision*.

Ethernet menggunakan sistem *contention* yang dikenal dengan *Carrier Sense Multiple Access with Collision Detection* atau CSMA/CD. Untuk melihat bagaimana sebenarnya teknologi ini bekerja, mari kita bagi namanya menjadi Carrier Sense—perangkat mendengarkan media akan adanya sinyal carrier, Multiple Access—perangkat berbagi pakai media, dan Collision Detection— perangkat dapat mendeteksi *collision*.

Ketika perangkat ingin mengirimkan data pada jaringan, pertama ia mendengarkan untuk memastikan tidak ada *node* lain yang mengirimkan data pada saat itu. Jika pada media tidak terdapat sinyal, ia akan mengirimkan data. Di sini jelas terdapat suatu masalah—adalah mungkin dua node akan mendeteksi media pada saat yang bersamaan dan mencoba mengirimkan data. Jika ini terjadi, akan muncul *collision* yang akan mengorupsi data yang dikirim.

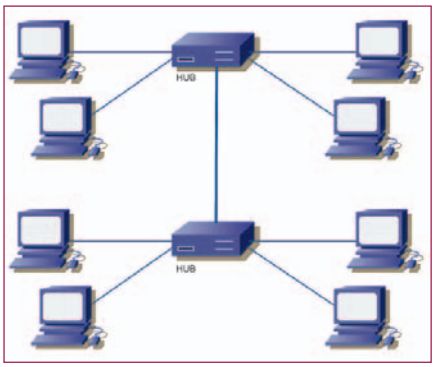
Ketika muncul collision, maka sistem perlu mengirimkan ulang data mereka. Agar collision tidak segera muncul, sistem yang terlibat dalam collision akan 'mundur' selama periode waktu tak tentu sebelum melakukan pengiriman ulang. Jika muncul collision lain, waktu mundur akan bertambah dan akan terus bertambah. Inilah yang menjadi alasan mengapa collision sangat mengganggu, terutama pada jaringan besar.

Untuk menghindari atau mengurangi *collision*, suatu jaringan harus dibagi ke dalam *collision domain* kecil. Ingat kembali ketika dihubungkan ke hub, semua sistem berada pada collision domain yang sama, di mana data mereka dapat bertabrakan satu sama lain. Collision domain dapat dibuat dengan menggunakan *bridge* dan *switch*. Ketika bridge ditambahkan ke jaringan, ia dapat memisahkan jaringan ke dalam beberapa collision domain. Ketika switch digunakan, lebih banyak lagi collision domain yang dapat dibuat—setiap port menjadi collision domain. Jika setiap sistem dihubungkan ke port switch-nya sendiri, maka tidak akan terjadi collision. Dengan demikian, akan lebih baik jika Anda mengganti hub dengan switch.

Yang menjadi alasan mengapa banyak perusahaan awalnya lebih memilih Token Ring daripada Ethernet adalah karena collison, dan dampak yang mereka rasakan terhadap kinerjanya. Dengan ditemukannya *bridging* (dan terutama *switching*) Ethernet menjadi lebih andal dan dengan sendirinya menjadi lebih populer. Ia menjadi teknologi LAN yang paling populer digunakan saat ini dan pangsa pasarnya semakin besar.

Bagaimana Topologi Ethernet?

Ketika awalnya ditetapkan, Ethernet bukanlah solusi yang sangat bersahabat. Implementasinya melibatkan penghubungan sistem ke satu kabel *coaxial* panjang yang dikenal dengan Thicknet atau 10Base5. Untuk menghubungkan ke Ethernet, sistem harus menggunakan kartu jaringan dengan *transceiver* luar yang benar-benar 'tergigit' ke dalam inti kabel. Hal ini dikenal dengan *vampire tap*. Transceiver tersebut kemudian dihubungkan ke kartu jaringan menggunakan kabel *Attachment Unit Interface*



Topologi *Star-wired bus.*

(AUI). Jenis konektor yang digunakan adalah konektor DIX, sesuai dengan nama perusahaan yang mengembangkannya.

Kemudian, Ethernet menggunakan kabel yang lebih tipis yang disebut 10Base2 atau Thinnet. Jaringan ini lebih fleksibel karena sistem dihubungkan langsung ke kabel coaxial menggunakan konektor BNC, bukan seperti yang terhubung ke bagian belakang televisi Anda.

Topologi yang digunakan adalah topologi bus, di mana sinyal dilewatkan sepanjang kabel. Pada bagian akhir, suatu perangkat yang disebut terminator menyerap sinyal tersebut, agar tidak dikembalikan lagi ke kabel dan menyebabkan *collision*.

Jaringan bus murni mungkin tidak begitu populer lagi sekarang ini, tetapi mereka masih menjadi dasar pengembangan jaringan Ethernet. Kabel pendek dan hub menggantikan satu kabel panjang, tetapi sinyal masih dapat berjalan ke semua sistem yang terhubung ke segmen. Ketika sistem terhubung ke hub, topologi yang digunakan biasanya topologi star. Oleh karena itu, Ethernet seringkali dikenal dengan *Star-wired bus*, seperti yang tampak pada gambar.

Bagaimana Pengalamatan Ethernet?

Alamat Ethernet juga dikenal dengan *Media Access Control* (MAC) atau alamat *hardware*. Alamat ini dihubungkan ke kartu jaringan dengan membakarnya pada *chip* ROM pada waktu produksi. Alamat Ethernet tidak hanya harus unik untuk setiap kartu (alamat ini juga berperan sebagai nomor seri), tetapi juga menunjukkan vendor yang memproduksi kartu tersebut.

Alamat Ethernet memiliki panjang 48-bit yang ditulis dalam bentuk heksadesimal. Sistem bilangan hex dikenal dengan Base16, karena terdapat 16 kemungkinan—dari 0 ke F. Pada tabel, Anda dapat melihat nilai setiap digit hex dalam desimal dan biner. Perlu dicatat bahwa nilai berapa pun di atas F tidak akan pernah *valid*, dan setiap alamat dalam hex terdiri dari 12 digit—setiap digit hex mewakili 4-bit.

Format alamat hardware yang Anda lihat bervariasi, tetapi secara umum ada dua format yang sering digunakan. Pada Windows, setiap byte alamat MAC dipisah oleh tanda hubung. Sebagai contoh: 00-50-BA-C3-0C-C7. Pada perangkat Cisco, alamat tersebut biasanya ditulis dalam tiga kelompok 4 digit hex. Sebagai contoh: 0050.BAC3.0CC7. Format yang digunakan tidak mengubah alamatnya karena itu hanya merupakan representasi. Namun, karena sekarang kita belajar Cisco, kita sebaiknya mengikuti format yang terakhir.

Yang penting dari alamat MAC adalah apa yang diidentifikasi olehnya. 24-bit pertama menunjukkan *Organizationally Unique Identifier* atau OUI vendor, sedangkan 24-bit terakhir menunjukkan kartu jaringan. Dari contoh sebelumnya, 0050BA menunjukkan vendor, dan C30CC7 menunjukkan nomor seri kartu. IEEE mengalokasikan OUI ke pabrikan. Perlu dicatat bahwa kesalahan tetap saja terjadi—ketika setiap kartu jaringan seharusnya memiliki alamat yang unik, banyak kesalahan yang dibuat selama proses produksi. Adanya alamat MAC yang sama dapat menyebabkan masalah pada jaringan, tetapi secara umum itu bukanlah masalah yang besar.

Apakah Full Duplex dan Half Duplex?

Ethernet awalnya didesain untuk bekerja sebagai sistem *half duplex*, yang berarti sistem hanya dapat mengirim atau menerima data, tetapi tidak dapat melakukan keduanya secara bersamaan. Pada kenyataannya, ini bukanlah bagian dari desain—ini sebenarnya fungsi akses media yang *contention-based*. Karena semua sistem berbagai pakai media, dan hanya satu sistem yang dapat mengirim data pada waktu yang ditentukan, maka half duplex menjadi kebutuhan.

Ketika sistem dihubungkan ke hub, atau terhubung ke bus Ethernet, mereka akan selalu berkomunikasi secara *half duplex*, walaupun ketika kartu jaringan mendukung *full duplex*. Untuk memungkinkan komunikasi full duplex, di mana sistem dapat mengirim dan menerima data secara bersamaan, harus menggunakan switch. Sistem memerlukan port switch sendiri agar dapat berkomunikasi secara full duplex.

Sebagai contoh, misalnya Anda menghubungkan satu sistem ke port switch dan menghubungkan hub ke port yang lain. Dengan menghubungkan hub ke switch membuat sejumlah komputer berada dalam *collision domain* yang sama. Namun, sistem yang terhubung ke hub tersebut hanya dapat berkomunikasi secara *half duplex*—lagi pula, mereka masih berada dalam jaringan tradisional yang berbagi pakai. Komunikasi dua sistem secara full duplex menggunakan *bandwidth* yang tersedia hanya terjadi ketika keduanya dihubungkan ke port masing-masing. Dengan demikian, sistem dapat mengirim dan menerima data secara bersamaan pada kecepatan maksimum NIC.

Ketika dihubungkan ke switch atau hub, kebanyakan sistem Ethernet sekarang ini dapat melakukan *autonegotiation*. Ketika sistem terhubung, port dan *interface* kartu jaringan bertukar *Fast Link Pulses* (FLP), yang membawa informasi mengenai kemampuan kartu. Sebagai contoh, jika Anda menghubungkan komputer dengan kartu Ethernet 10/100 Mbps ke port switch 10Mbps, mereka secara otomatis sepakat pada kecepatan *full duplex* 10 Mbps Perlu dicatat bahwa tidak semua kartu jaringan dan switch atau hub dapat melakukan autonegotiation, terutama model lama.

Sekian dulu perjumpaan kita kali ini. Pada artikel berikutnya kita akan kembali membahas Ethernet, dengan melihat jenis frame, teknologi Ethernet, dan lainnya. Sampai jumpa lagi dan sukses selalu. ■

Tabel Bilangan

Heksadesimal	Desimal	Biner
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Surga Download Peer-to-Peer

Hayri

Jika *search engine* sudah tidak lagi memberikan apa yang Anda cari, mungkin program *peer-to-peer client* dapat sedikit membantu.

Bisa mendapatkan apa yang Anda cari di Internet tentu saja siapa yang tidak senang. Bisa mendapatkan film yang ingin Anda tonton, mendapatkan lagu kesukaan yang ingin didengar, bisa mendapatkan program yang Anda butuhkan, semua itu mungkin merupakan impian dari semua pengguna Internet.

Biasanya dalam mencari sesuatu, kali pertama yang dikunjungi mungkin adalah sebuah situs *search engine*. Di sana Anda dapat membubuhkan nama file, judul film, judul lagu, atau apa saja yang berhubungan dengan yang Anda cari. Hasilnya, tergantung pada engine dan algoritma dari search engine yang Anda gunakan tersebut. Jika Anda beruntung, semua yang diinginkan dapat Anda miliki.

Namun menggunakan search engine untuk mencari sesuatu, Anda sering kali akan dihadapkan dengan banyak sekali jalan pilihan. Begitu Anda masukkan kata-kata kunci, dalam sekejap keluar ratusan *link* yang menurut search engine, sesuai dengan apa yang Anda cari. Padahal belum tentu sesuai untuk Anda. Proses pencarian kedua pun segera berlangsung. Anda harus mengubek-ubek isi dari setiap link yang ditampilkan. Terkadang Anda harus menggali dan terus menelusuri link-link yang ada pada sebuah halaman web untuk mendapatkan sebuah file, misalnya. Belum lagi kalau ternyata file yang Anda cari tidak ada di halaman web tersebut, padahal Anda sudah mengubek-ubeknya hingga dalam.

Jika Anda memang senang mencari file-file gambar, aplikasi, lagu, ebook, dan banyak lagi untuk di-*download*, ada alternatif kedua untuk memudahkan Anda mendapatkan apa yang dicari. Anda bisa menggunakan bantuan program dan jaringan peer-to-peer (P2P).

Apakah Jaringan Peer-to-Peer?

Mungkin Anda sudah sering mendengar jaringan bersistem *client/server*, di mana dalam sistem jaringan client/server ini terdapat sekelompok PC yang bertindak sebagai client dan sekelompok lagi bertindak sebagai server. Server berfungsi melayani setiap keperluan dan permintaan dari client-nya. Begitu ada masalah dengan server-server tersebut, seluruh client tidak dapat berbuat apa-apa lagi. Tetapi, itu tidak akan terjadi pada jaringan *peer-to-peer*.

Jaringan peer-to-peer adalah sebuah sistem jaringan yang tidak memiliki perbedaan derajat atau tingkatan dari setiap anggotanya. Semua anggota dapat saling melayani, dapat saling berkomunikasi satu dengan lainnya dan jika salah satunya bermasalah, maka tidak akan berpengaruh ke seluruh jaringan.

Selain itu, jaringan P2P juga memiliki kelebihan dalam melakukan distribusi file. Pendistribusian file dapat menjadi sangat luas karena file tidak hanya dapat didistribusikan dari satu titik saja, melainkan dari banyak titik distribusi. Maka itu, search engine yang menggunakan jaringan peer-to-peer menjadi sangat *powerful* dan sangat reliabel dalam mencari sesuatu. Berdasarkan atas teori tersebut, maka saat ini banyak sekali bermunculan jaringan peer-to-peer yang memungkinkan Anda untuk mencari sesuatu yang tidak Anda temui di server-server web hasil pencarian *search engine* biasa. Tetapi, Anda mungkin akan menemui apa yang Anda cari di dalam PC pribadi seseorang di luar sana yang juga tergabung di dalam jaringan peer-to-peer.

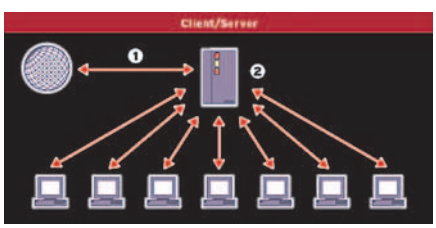
Apakah Tersedia Pilihan untuk Menggunakan Jaringan P2P?

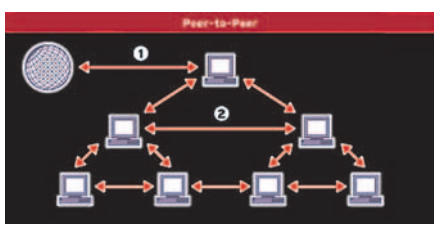
Jaringan P2P yang memungkinkan Anda mencari sesuatu di dalamnya mulai *booming* sejak tahun 1999, di mana seiring dengan meledaknya penggunaan Internet di mana-mana. Sejak itu, banyak sekali jenis jaringan P2P yang dapat Anda pilih dan gunakan sebagai *search engine* dan juga media tukar-menukar file, mulai dari MP3 sampai *software-software* aplikasi. Berikut ini adalah beberapa jaringan P2P yang pernah atau masih ada sampai sekarang.

Napster

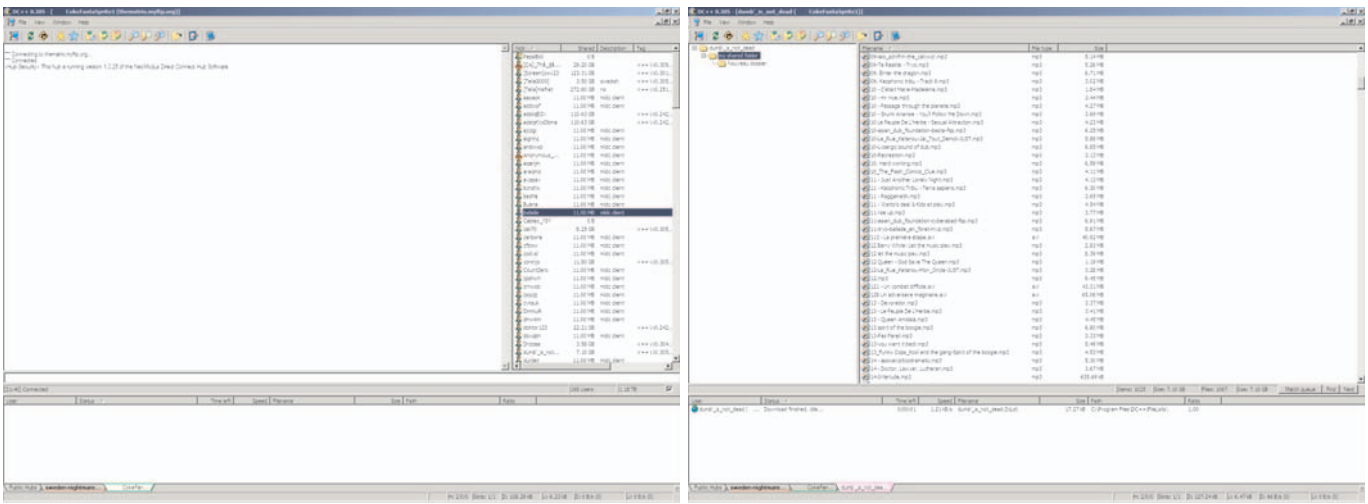
Napster kali pertama ditemukan oleh seorang remaja 18 tahun bernama **Shawn Fanning** pada tahun 1999. Tujuan utamanya membuat Napster adalah agar ia lebih mudah untuk dapat tukar-menukar file khususnya koleksi MP3-nya dengan orang lain. Karena MP3 sedang digilai di mana-mana saat itu dan ia sering kali kesulitan untuk menemukan apa yang ia cari, maka muncullah ide untuk membuat sebuah program yang dapat membangun jaringan antarpemilik file-file MP3 yang dapat terhubung secara langsung (*peer-to-peer*).

Pada saat pembuatannya, ada tiga tujuan utama yang ingin dicapainya dalam membuat program tersebut, yaitu membuat MP3 Search Engine agar ia mudah mencari MP3 yang dikehendakinya. Setelah ditemukan, file tersebut harus bisa di-download, maka itu ia menyertakan kemampuan file sharing pada programnya. Selain itu, untuk





Kedua jenis jaringan ini memiliki kelebihan dan kekurangannya masing-masing. Namun dalam hal pendistribusian file untuk di bagi dengan pengguna lain, jaringan model *peer-to-peer* tampaknya memang lebih unggul dalam hal skalabilitas.



Dengan menggunakan DC++, Anda dapat terkoneksi ke banyak sekali publik hub yang di dalamnya terdapat banyak para pengguna yang juga membuka sebuah folder-nya untuk diakses oleh umum.

dapat saling berkomunikasi antarsesama ia juga menyertakan kemampuan *Internet Chat Relay* (IRC) pada programnya. Setelah semuanya ada, maka jadilah program peer-to-peer Napster yang mungkin sempat juga Anda gunakan beberapa tahun lalu.

Untuk dapat terkoneksi ke dalam jaringan Napster, Anda membutuhkan sebuah program klien Napster dan tentunya koneksi Internet. Begitu program klien diinstal, maka Anda akan diminta untuk menentukan folder mana yang ingin Anda jadikan sebagai tempat penampungan sekaligus *share folder* yang dapat diakses oleh para pengguna Napster lain. Selanjutnya begitu Anda menjalankan program klien Napster pada PC Anda, Napster akan melakukan pengecekan apakah PC Anda terkoneksi ke Internet atau tidak. Jika koneksi Internet sudah tersedia, maka program tersebut akan menghubungi server pusat Napster. Jika server berhasil dihubungi, maka PC Anda tersebut akan langsung tergabung (*logon*) ke dalam server tersebut.

Tujuan dari adanya server pusat ini bukanlah difungsikan sebagai tempat penampungan file MP3 yang Anda inginkan, melainkan hanya sebagai pengindeks saja. Server ini akan mendata *user-user* yang sedang menjalankan program klien Napster pada saat itu dan file apa saja yang di-share oleh mereka. Semua masuk ke dalam *indexing* di server pusat. Ketika ada pengguna yang memasukkan kata kunci dari file yang dicari, maka program klien Napster akan mencarinya terlebih dahulu di server pusat, apakah ada pengguna yang memiliki file tersebut. Jika server menemukan apa yang Anda cari, maka hasilnya akan diberitahukan kepada Anda melalui program klien itu juga. Hasilnya berupa sederetan daftar file yang ada pada PC lain yang berhasil diindeks oleh server tersebut.

Ketika Anda memutuskan untuk mendownload file yang ditemukan, maka program klien Napster Anda akan membangun sebuah koneksi dengan PC tempat file tersebut berada. Mungkin saja PC tersebut ada di sebelah rumah Anda, atau mungkin juga berada di belahan lain di bumi ini. Setelah koneksi berhasil terjalin, file akan mulai di-download. Setelah selesai di-download, maka PC tadi akan memutuskan koneksi dengan PC Anda. Begitu seterusnya proses ini berulang.

Selain Windows, para pengguna Linux, MAC, BEOS, OS/2, dan banyak lagi *operating system* lain juga dapat menikmati jaringan Napster dengan adanya OpenNap, yaitu protokol Napster yang bersifat *open source*. Untuk aplikasi kliennya digunakan Lopster, Knapster, macster, dan banyak lagi.

Direct Connect

Direct Connect adalah sebuah jaringan *peer-to-peer* yang tergolong sebagai jaringan *semi-centralized*. Maksudnya semi-centralized adalah beberapa penggunanya harus menjalankan sebuah 'hub' yang mana hub ini adalah sebuah server yang berfungsi sebagai pengatur pengindeksan pengguna, pengindeksan file, dan sebagai *chat server*.

Ketika selesai menginstal dan menjalankan aplikasi Direct Connect di PC Anda lalu terhubung ke Internet, Anda akan mendapatkan PC Anda sudah menjadi bagian dari sebuah hub yang ada di jaringan Direct Connect. Anda akan melihat sebuah *list* berisikan user-user lain termasuk juga Anda, seperti halnya sebuah kanal *chatting*. Di list ini kemudian Anda dapat menjelajah file-file yang di-*share* oleh para pengguna tersebut. Anda dapat menjelajah 'isi perut' dari masing-masing individu dan kemudian mengambil yang Anda perlukan atau dapat juga melakukan pencarian file ke seluruh pengguna di dalam hub tersebut.

Sebuah hub dapat juga membentuk sebuah jaringan hub yang dapat juga Anda *ubek-ubek*. Untuk dapat masuk ke dalam sebuah jaringan hub, Anda akan melewati sebuah proses *polling*, jadi saat kali pertama masuk Anda akan ditempatkan di sebuah server yang memang masih lowong. Setiap hub memiliki peraturannya masing-masing yang juga ditentukan oleh para penggunanya. Misalnya, sebuah hub hanya memfokuskan diri pada sharing file-file MP3 dan Divx, atau peraturan yang paling umum adalah sebuah hub menetapkan nilai minimum dari banyaknya file yang harus Anda share. Jika Anda memiliki misalnya 2 GB file yang di-share

ke jaringan Direct Connect, maka Anda baru diperbolehkan bergabung dalam hub tersebut. Besarnya ketentuan ini bervariasi pada masing-masing hub. Pengguna yang tidak memiliki file untuk di-share akan sangat sulit untuk bergabung ke hub-hub tersebut. Ketentuan-ketentuan ini menjadikan Direct Connect sebagai jaringan *file sharing* yang paling besar karena tidak sedikit pula pengguna-pengguna yang melakukan sharing file hingga sebesar ratusan Gigabyte.

Yang paling mengagumkan, Direct Connect dibuat dengan menggunakan bahasa Visual Basic dengan tetap mempertahankan kemudahan dan kerapian *interface*-nya. Di dalam Direct Connect disisipkan iklan terselubung pada awal versinya, namun setelah diprotes oleh banyak pihak, maka pada versi-versi selanjutnya Anda dapat menikmati jaringan ini dengan bebas iklan.

Aplikasi-aplikasi yang dapat digunakan sebagai client Direct Connect adalah DC++ untuk pengguna yang mengandalkan Windows sebagai *operating system*-nya, MLDonkey juga dapat Anda gunakan untuk bergabung dalam jaringan Direct Connect jika Anda menggunakan Linux. Salah satu aplikasi yang memiliki kelebihan khusus adalah C-Keen yang dapat melakukan multi-download dari beberapa penguna berbeda secara berbarengan dalam satu waktu.

FastTrack

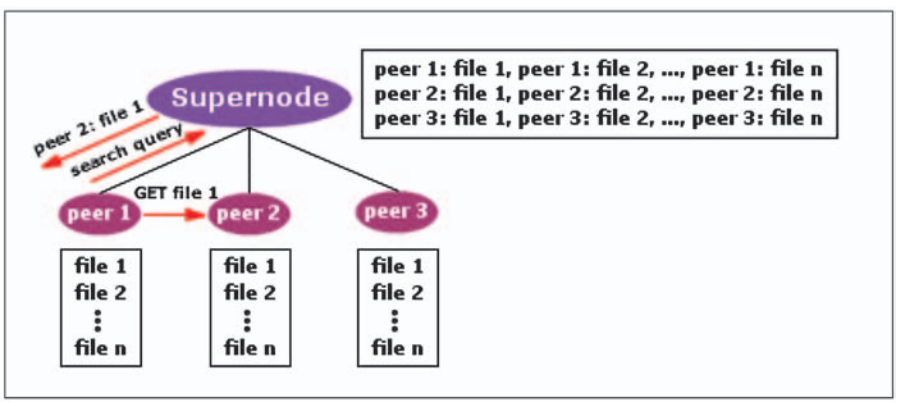
Jaringan *FastTrack* merupakan jaringan yang digunakan oleh program-program *client peer-to-peer*, seperti Kazza, Grokster, dan iMesh.

Sejak awal tahun 2003, jaringan FastTrack merupakan jaringan peer-to-peer yang paling populer yang banyak digunakan orang untuk bertukar-tukar file baik MP3, Divx, jpg, maupun yang lainnya. Penggunanya yang melakukan transaksi MP3 bahkan melebihi pengguna Napster pada masa puncaknya. Hal ini tentu disebabkan oleh adanya fitur dari FastTrack yang membuat penggunanya terbuai, yaitu kemampuannya untuk melakukan resume ketika proses *download* terputus. Satu lagi yang menjadikan Kazaa dan iMesh populer adalah kemampuannya melakukan download sebuah file dari beberapa pengguna secara simultan.

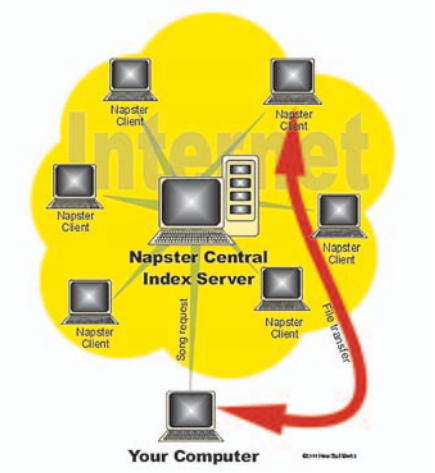
Jadi jika ada lima pengguna yang memiliki file yang sama, program klien FastTrack dapat membagi-bagi file tersebut menjadi lima bagian yang di-download dari kelima pengguna tersebut secara bersamaan. Jadi, paling tidak prosesnya akan lebih cepat.

Jaringan FastTrack ini juga disebut-sebut sebagai jaringan P2P generasi kedua, di mana sebenarnya jaringan ini menggunakan protokol Gnutella dengan disisipi oleh fitur tambahan untuk meningkatkan faktor skalabilitasnya, yaitu sistem Supernodes.

Supernodes adalah semacam server indeks yang dibangun di sebuah PC yang menjalankan aplikasi client juga. Namun PC yang dapat terpilih sebagai Supernodes adalah PC yang memiliki tenaga *processing* yang besar dan koneksi Internet yang cepat. Supernodes terpilih secara otomatis dan akan menjadi



Kunci daripada keberhasilan Kazaa, iMesh, dan kawan-kawan sejenisnya adalah dengan adanya fitur Supernodes yang bertindak sebagai *server* sentral, namun tetap berjalan seperti halnya *client-client* yang lain.



Koneksi *peer-to-peer* dapat dibangun dengan bantuan *server* sentral yang merupakan kunci dari jaringan Napster.

indeks server sementara bagi client-client lain yang lebih lambat koneksinya.

Untuk dapat membangun koneksi kali pertama, program client seperti Kazaa misalnya memiliki daftar alamat IP Supernodes yang sudah tetap yang disisipkan dalam programnya. Setelah terkoneksi ke Supernodes tetap tersebut, client tersebut meminta daftar Supernodes lain yang sedang aktif untuk digunakan pada koneksi selanjutnya. Kemudian program client akan memilih salah satu Supernodes sebagai '*upstream*' dan mengirimkan semua informasi file yang dimilikinya. Kemudian Supernodes yang satu dengan yang lainnya akan saling berkomunikasi untuk bertukar informasi dan melayani pencarian dari client.

Setelah Supernodes memberikan informasi yang dibutuhkan client-nya, koneksi peer-to-peer antara client yang satu dengan yang lain akan terjalin dan file transfer dimulai. Transfer file ini dilakukan dengan menggunakan protokol HTTP. Jadi sebenarnya ada dua lapisan jaringan pada jaringan FastTrack ini, yaitu jaringan Supernodes dan jaringan antarpenggunanya itu sendiri.

BitTorrent

Sistem jaringan *peer-to-peer* yang satu ini memiliki cara kerja agak unik dari yang lainnya. Meskipun jaringan ini memang membangun koneksi langsung antara para penggunanya, namun sebuah *server* sentral masih tetap dibutuhkan untuk mengatur kelan-

carannya (sistem *semi-centralized*). Server sentral ini sering disebut sebagai *tracker*. Sebuah server tracker berfungsi untuk mencatat dan mengoordinasikan setiap *action* yang terjadi pada semua *peer* atau *host* yang ada. Tracker tidak menyimpan data atau indeks mengenai file-file apa saja yang di-*share* oleh pengguna. Maka dari itu, sebuah server tracker dapat melayani banyak pengguna meskipun *bandwidth*-nya terbatas.

Filosofi dari jaringan *BitTorrent* ini adalah setiap pengguna harus melakukan *upload* pada saat yang bersamaan dengan proses download yang sedang mereka lakukan terhadap sebuah file. Dengan adanya sistem ini, maka bandwidth yang Anda miliki akan digunakan secara efisien karena proses upload dan download memiliki porsi dan waktu yang sama.

Dengan adanya desain ini pula, BitTorrent memiliki karakteristik yang sedikit berbeda dari sistem jaringan lain. Jika jumlah pengguna yang menginginkan sebuah file semakin meningkat, itu berarti semakin cepat pula performa download yang dapat Anda rasakan. Hal ini sama sekali berkebalikan dengan sistem jaringan peer-to-peer yang lain, jika banyak yang men-download sebuah file dari sebuah PC pasti performa download akan melambat.

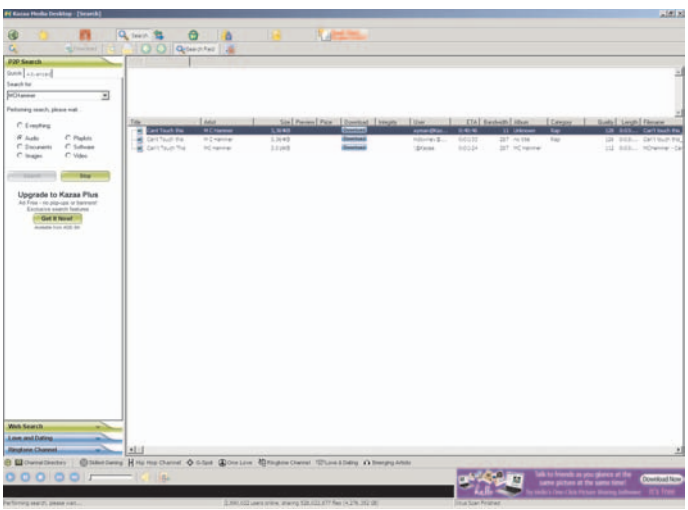
Analogi untuk sistem jaringan yang cukup rumit ini adalah seperti sebuah grup yang duduk dalam sebuah meja dan sedang berusaha mendapatkan sebuah salinan buku yang tercerai-berai. Masing-masing dari mereka memegang sepotongan halaman salinan buku. Si A memberitahu kepada grup bahwa ia memiliki beberapa lembar halaman 1-10, 25, 31-59 dan 75. Si B menginformasikan ke grup bahwa ia memegang halaman 11-24, 60-70 dan 74. Si C, D, dan E juga memegang halaman-halaman buku yang tidak komplit dan menginformasikannya ke seluruh anggota.

Untuk mendapatkan sebuah buku yang utuh bagi masing-masing orang, maka mereka melakukan pertukaran. Si A dapat me-*request* apa yang tidak dimilikinya kepada si B, C, D, dan E, begitu pula sebaliknya mereka juga dapat melakukan request apa yang tidak dimilikinya ke si A dan memberikan kepadanya. Pertukaran terus berlangsung di meja dan semua dapat saling berkomunikasi melakukan pertukaran hingga semua bagian tersebar dengan baik.

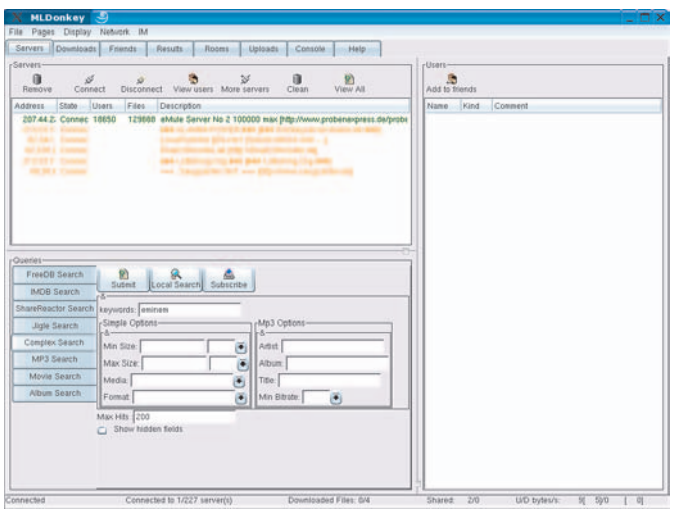
Ada seorang lagi yang memiliki sebuah salinan yang lengkap, anggap saja bernama S. Ia akan bertindak sebagai pelengkap bagi halaman-halaman yang tidak dimiliki oleh A, B, C, D, dan E. Ia tidak meminta apapun dari grup, tetapi akan merespon permintaan halaman yang tidak ada di grup tersebut. Ketika ada seorang anggota yang baru masuk, ia akan lebih dulu melakukan permintaan halaman ke si S. Setelah dapat beberapa, baru anggota-anggota lain juga akan dihubunginya untuk dimintai. Setelah semuanya mendapat salinan lengkapnya, maka mereka juga akan melakukan hal yang sama dengan si S. Dalam terminologi BitTorrent si S disebut sebagai *Seed*. Jaringan BitTorrent memiliki sistem kerja yang mirip dengan analogi ini.

Dalam jaringan BitTorrent, masing-masing pengguna dapat membagi file-file mereka dengan sesama pengguna lain tanpa harus memberikan file tersebut dalam keadaan utuh. Si peminta akan melakukan pencarian ke pengguna-pengguna lain yang juga memiliki file tersebut secara otomatis.

Dengan adanya sistem ini, penggunaan *bandwidth* Anda tentu akan lebih efisien karena tidak boros. BitTorrent akan sangat efektif jika digunakan untuk mencari file-file baru atau yang banyak dicari orang karena semakin banyak yang mencari, semakin tersedia banyak pula sumbernya. Yang bertugas mendata dan mengoordinasikan semua ini adalah *server tracker*. Semua informasi mengenai pengguna ada di dalam tracker. Masing-masing pengguna akan melakukan *update* informasi ke tracker ini untuk mengetahui perkembangan jaringan. Beberapa aplikasi yang dapat digunakan untuk bergabung ke jaringan BitTorrent adalah Bit Spirit, BT++, Shareaza, TorrentStorm, dan banyak lagi.



Menggunakan Kazaa, Anda akan mendapatkan cukup banyak fitur seperti *Family Filter* untuk memblokir hasil-hasil *search* yang porno dan disediakan juga *Bullguard Antivirus* untuk menjaga kenyamanan Anda ber-*download* ria.



Aplikasi MLDonkey melayani para pengguna Linux untuk mengoneksikan diri ke jaringan eDonkey. Namun tidak hanya eDonkey saja, versi-versi terbarunya biasanya menyertakan fasilitas untuk terkoneksi ke jaringan-jaringan lain seperti BitTorrent, Direct Connect, dan banyak lagi.

eDonkey2000 dan OverNet

Jaringan eDonkey2000 versi yang terbaru tidak hanya menggunakan jaringan eDonkey saja, namun juga menggunakan jaringan OverNet. Jaringan eDonkey dan OverNet menggunakan protokol yang sama dalam melakukan file sharing. Namun, perbedaan terbesar keduanya adalah eDonkey menggunakan sistem sentralisasi server, sedangkan OverNet menggunakan sistem *serverless*.

Meski demikian, jaringan eDonkey tidak memiliki sebuah server sentral yang tetap yang khusus dibuat. Server-server sentral ini adalah PC para pengguna yang dengan sukarela menjalankan aplikasi server eDonkey2000. Jadi server-server tersebut bisa di mana saja di belahan dunia ini dan dengan alamat IP mana saja. Pertama kita akan membahas sistem kerja eDonkey lebih dulu.

Sama seperti kebanyakan jaringan peer-to-peer yang menggunakan bantuan server sentral, eDonkey juga mengandalkan peran serta server-server sentral untuk melakukan indeksing pengguna dan file-file yang mereka share. Namun, yang membedakannya adalah kemampuannya untuk melakukan pencarian ke beberapa server sekaligus. Anda akan jarang sekali kembali dengan tangan hampa ketika melakukan pencarian karena begitu Anda terkoneksi ke beberapa server, maka secara otomatis Anda akan berada di dalam jaringan yang sangat luas. Dan Anda bisa men-download dari manapun file tersebut ditemukan.

eDonkey memungkinkan Anda men-download file berjenis apapun, tidak hanya MP3 saja. Selain itu, Anda juga akan menikmati fasilitas resume download jika suatu saat proses download Anda harus terputus. Proses resumenya pun tidak harus menggunakan server yang sama seperti sebelumnya. Proses transfer juga dapat dilakukan dari beberapa server sekaligus, sehingga transfer data menjadi lebih cepat. Sebuah file juga dapat di-*upload* dan diambil oleh pengguna lain sementara Anda sedang men-download-nya. Jadi, *bandwidth* menjadi lebih efektif.

Tidak hanya itu, aplikasi eDonkey dapat berjalan dengan tidak terikat pada suatu port tertentu, jadi Anda dapat menggunakan port komunikasi berapa saja untuk terkoneksi ke jaringan eDonkey. Masih ada lagi, aplikasi eDonkey dapat Anda miliki tanpa mengeluarkan biaya sedikit pun dan tidak disertai dengan iklan-iklan terselubung. Pengguna Linux, Windows, bahkan Mac juga dapat menikmati kehebatan jaringan eDonkey.

OverNet juga merupakan jaringan peer-to-peer yang cukup hebat. Dengan menggunakan protokol yang sama dengan apa yang digunakan eDonkey, jaringan ini dapat menghilangkan peran jaringan client server sama sekali. Tidak ada level client dan level server di dalam jaringan ini. Semua client yang menjalankan aplikasi OverNet disebut sebagai *node* dan memiliki derajat yang sama. Setiap node memiliki seperangkat informasi mengenai node-node lain yang ada di dalam jaringan. Untuk bergabung ke dalam jaringan, setiap node harus mengetahui alamat IP dan port yang digunakan oleh node lain. Setelah alamat IP dan nomor port diketahui, maka node tersebut bisa melakukan proses *bootstrap* ke node-node lain.

Proses bootstrap adalah proses yang kali pertama dilakukan untuk benar-benar bergabung ke jaringan OverNet. Setelah sebuah node mengetahui alamat IP node lain, ia bisa meminta informasi node-node lain yang diketahuinya. Setelah mendapatkan informasi node lain, maka node pertama bisa melakukan koneksi ke node tersebut dan begitu seterusnya hingga semua node terhubung. Proses ini sebenarnya sangat sederhana dan tidak memakan waktu lama.

Untuk node-node lain di dalam jaringan mudah dikenali, maka setiap node diberi pengalamatan khusus. Pengalamatan ini dilakukan secara acak dan terdiri dari 128-bit ID. Ketika sebuah node ingin mengirimkan informasi mengenai file-file yang di-share-nya, maka informasi ini juga disertai dengan 128-bit ID. Node yang memiliki nomor ID paling dekat dengan ID file tersebut adalah yang paling bertanggung jawab atas sharing file itu. Proses pencarian dan publikasi file sharing akan banyak sekali mengandalkan nomor ID ini.

Seperti halnya eDonkey, aplikasi OverNet juga tersedia untuk para pengguna Linux dan Mac OS selain daripada pengguna Windows. Anda dapat men-download-nya tanpa keluar biaya apapun di situs mereka *www.edonkey2000.com*.

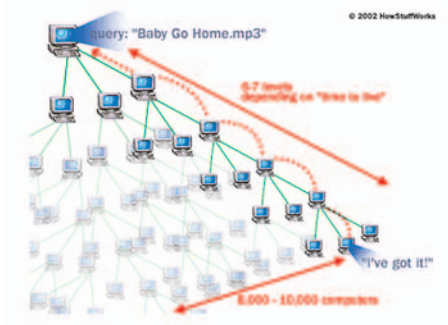
Gnutella

Setelah kematian Napster karena kalah di pengadilan kasus pelanggaran hak cipta, hampir 60 juta penggunanya bagai kehilangan nafas. Mereka tidak bisa lagi melakukan *sharing* dan saling bertukar file MP3 karena nadi utama dari jaringan Napster, yaitu *server* pusat database untuk menyimpan indeks lagu dan pengguna, di-*shut down* oleh pengadilan. Tetapi 'kelaparan' para pengguna ini tidak terlalu lama karena kebanyakan pengguna Napster langsung menyerbu jaringan *peer-to-peer* lainnya pada saat itu, yaitu Gnutella.

Ada persamaan dan perbedaan antara kedua sistem jaringan ini. Namun, yang pasti Gnutella dibuat sedemikian rupa sehingga kesalahan pada Napster tidak terulang lagi, yaitu memperbaiki masalahnya dengan hukum hak cipta. Ada beberapa kesamaan antara sistem Napster dengan Gnutella. Mereka sama-sama membiarkan file-file penggunanya diakses oleh pengguna lain, sehingga membentuk jaringan peer-to-peer.

Selain itu, untuk terkoneksi ke jaringan, mereka sama-sama mengharuskan penggunanya untuk menjalankan aplikasi tertentu. Perbedaan yang paling mendasar adalah tidak adanya server pusat database pada jaringan Gnutella seperti halnya pada jaringan Napster. Selain itu, Gnutella memiliki banyak sekali aplikasi client yang tersedia untuk digunakan. Dengan adanya fitur-fitur pada Gnutella ini, maka tidak akan mungkin bagi pengadilan sederhana untuk dapat mematikan jaringan besar ini.

Proses dan kerja jaringan Gnutella untuk melayani Anda melakukan pencarian sebenarnya cukup sederhana. Ketika Anda memasukkan kata kunci yang bersangkutan dengan file yang dicari, program client Gnutella akan menghubungi paling tidak satu PC lain yang juga menjalankan aplikasi client. Untuk menghubungi salah satu PC ter-



Arsitektur jaringan Gnutella memang membebaskan masalah yang dialami oleh Napster, tetapi masih belum cukup hebat untuk melayani para penggunanya.

sebut, Anda harus memiliki alamat IP dari PC lain, atau paling tidak program client Anda sudah memasukkannya ke dalam program tersebut. Setelah berhasil menghubungi salah satu PC tetangga, data pencarian Anda akan dikirim ke PC tersebut. Jika PC tersebut memiliki file sesuai dengan apa yang Anda cari, maka data file yang Anda cari akan dikirim balik ke PC Anda. Bersamaan dengan itu, PC tetangga tersebut juga mengirimkan *request* pencarian ke PC PC tetangganya yang ia tahu, sehingga proses akan terus berulang.

Namun pengulangan proses ini ada batasnya, yaitu dibatasi dengan adanya parameter *Time To Live* (TTL) dari proses request data ke tetangga-tetangganya. Ketika Sebuah PC melakukan pencarian ke satu PC tetangga, maka counter TTL akan berkurang satu. Begitu pula jika PC tetangga itu meneruskan pencarian ke tetangganya lagi, maka nilai TTL dari sesi pencarian itu akan berkurang lagi.

Nilai TTL ini akan berakhir setelah proses pencarian mencapai enam atau tujuh tingkatan ke bawah secara langsung. Proses pencarian ke tetangga akan berakhir ketika nilai TTL ini mencapai 0. Setiap PC yang dihubungi dalam pencarian ini juga memiliki nilai TTL yang sama, jadi masing-masing PC tadi juga bisa menghubungi tetangga-tetangganya yang lain hingga enam sampai tujuh tingkat lagi.

Sistem jaringan seperti ini memiliki keuntungan juga kerugian. Keuntungannya Anda bisa tergabung ke dalam jaringan Gnutella setiap saat, asalkan terhubung ke salah satu PC tetangga. Tidak akan ada pengadilan yang dapat menutup jaringan ini seperti halnya Napster karena sistem ini tidak berpusat pada sebuah server pusat.

Kekurangan dari jaringan ini adalah pada waktu pencariannya yang cukup lama. Untuk mencapai tujuh tingkat ke dalam dan mendapatkan informasi dari kurang lebih 8000 PC, membutuhkan waktu sekitar 1 hingga 2 menit. Selain itu, meneruskan pencarian dari tetangga-tetangga Anda adalah pemborosan *bandwidth* dan tenaga *processing* menurut sebagian orang.

Aplikasi client untuk Gnutella cukup banyak tersedia sebagai pilihan untuk Anda. Ini juga merupakan suatu kelebihan dari sistem jaringan ini. Jika Anda pernah mendengar Morpheus atau WinMX, aplikasi itu juga menggunakan sistem Gnutella. Selain itu, ada XoloX, Lime Wire, BearShare, dan Gnucleus. Masing-masing aplikasi memiliki kelebihan tersendiri.

Bagaimana dengan Keamanannya?

Tergabung dalam sebuah jaringan publik, tentu saja menimbulkan masalah baru bagi keamanan PC Anda. Terkoneksi ke jaringan publik seperti Internet dengan dilapisi *firewall* dan sistem keamanan lain saja belum tentu menjamin PC Anda akan aman. Apalagi sampai membuka lebar-lebar isi harddisk Anda untuk diakses dari luar, tentu membuka celah baru pula bagi PC Anda maupun jaringan lokal Anda. Banyak sekali peluang para *hacker* untuk membuka *backdoor* ke jaringan Anda melalui jaringan P2P ini. Belum lagi virus, *worm, adware, spamware*, dan banyak lagi gangguan-gangguan lain yang mungkin akan bercokol di PC Anda. Yang lebih menakutkan, data berharga Anda, kode *software* yang bernilai jutaan dolar, *password-password* e-banking Anda dapat pula disalurkan ke luar lewat jaringan ini.

Selain itu, menggunakan jaringan P2P juga akan membuat jaringan dan koneksi Internet Anda menjadi padat. File-file multimedia yang berukuran besar, file aplikasi, dan berbagai jenis file lainnya menjadi lebih sering keluar-masuk, baik di jaringan lokal maupun ke Internet. Belum lagi jika Anda menggunakan jaringan P2P yang melewatkan proses pencarian pengguna lain melalui PC dan jaringan Anda. *Bandwidth* Anda untuk bekerja menjadi semakin sempit saja.

Masalah keamanan memang merupakan masalah yang sampai saat ini belum dapat dipecahkan sepenuhnya. Anda hanya dapat mengandalkan 'kepercayaan' dalam hal ini. Anda harus mencoba untuk mempercayai para pengguna lain, harus mempercayai aplikasi yang Anda gunakan, dan banyak lagi. Salah satu metode yang baru-baru ini diusulkan untuk menjamin kepercayaan Anda adalah dengan membuat semacam penilaian terhadap tingkat kepercayaan seorang pengguna jaringan.

Setiap pengguna akan diberikan semacam tanda tangan digital (*digital signature*), yang kemudian digital signature tersebut memiliki semacam nilai kepercayaan. Nilai tersebut memiliki tingkatan-tingkatan katakanlah dari 0 sampai 20. Seseorang pengguna akan dapat dilihat tingkat kepercayaannya dari nilai tersebut. Semakin tinggi nilainya, maka akan semakin dipercaya pengguna tersebut. Metode ini mungkin sudah diterapkan di beberapa aplikasi P2P versi terbaru. Mungkin metode ini dapat sedikit menolong Anda menjaga keamanan PC Anda dalam menggunakan jaringan P2P.

Bersiaplah Memperbesar Bandwidth

Jika Anda sudah merasakan betapa nikmatnya mendapatkan apa yang Anda cari dengan menggunakan bantuan jaringan *peer-to-peer*, mungkin Anda akan menjadi 'pecandu' *download*. Rasanya ingin lagi dan lagi mencari sesuatu yang menarik untuk Anda di jaringan tersebut. Namun ada juga yang terus dan terus menghalangi Anda, yaitu kebutuhan akan *bandwidth* Internet. Jika Anda melakukan proses download, maka bandwidth untuk yang lain tentu akan berkurang jatahnya. Semakin sering, semakin banyak, maka semakin Anda merasa kehausan akan bandwidth yang besar. Maka itu, pergunakanlah jaringan peer-to-peer dengan bijaksana jika Anda tidak ingin kekurangan bandwidth terus menerus. ■

Monitor Jaringan dengan Protokol SNMP

Hayri

Jaringan yang ideal adalah jaringan yang didesain, diimplementasikan, dan dimonitor dengan baik. Protokol SNMP adalah salah satu kuncinya untuk dapat memonitor jaringan dengan baik.

Jaringan komputer dari hari ke hari semakin meluas saja penggunaannya. Di dalam perusahaan jaringan komputer semakin diperluas untuk melayani kebutuhan bisnis yang semakin cepat denyutnya. Di sekolah-sekolah, semua pelajar ingin diperkenalkan dengan komputer sejak dini, maka itu jaringannya semakin dikembangkan. Di pemerintahan semua sistem administrasi juga semakin bergantung kepada adanya jaringan komputer. Jaringan dari hari ke hari juga semakin kompleks saja.

Sebuah jaringan yang kompleks yang sangat diandalkan untuk dapat memudahkan pekerjaan tidak hanya sekadar '*up and running*' saja, namun juga harus bekerja secara optimal. Mengelola dan merawat sebuah jaringan komputer agar bekerja optimal tidaklah mudah. Banyak sekali opini dari orang-orang di luar sana yang sering mengatakan bahwa mengelola jaringan perlu sedikit unsur 'keajaiban' karena relatif sangat sulit untuk mengetahui apa sumber dari sebuah masalah ketika itu terjadi. Opini itu mungkin saja muncul dari para administrator jaringan yang tidak memiliki sistem monitoring yang baik. Jika memang tersedia sistem monitoring jaringan yang responsif dan dapat dipercaya pada jaringan mereka, mungkin opini itu tidak lagi ada di benak mereka.

Sebenarnya cukup banyak cara untuk melakukan monitoring jaringan, namun yang tampaknya sudah menjadi standar secara *de facto* untuk melakukan monitoring adalah dengan menggunakan protokol *Simple Network Management Protocol* (SNMP).

Apa Sebenarnya Protokol SNMP?

Simple Network Management Protocol adalah sebuah protokol *Application layer* pada standar 7 OSI layer dan merupakan bagian dari protokol TCP/IP yang banyak digunakan saat ini. Protokol ini biasanya digunakan untuk mengatur pertukaran database informasi yang menyangkut sistem manajemen dari sebuah perangkat jaringan. Dengan adanya pertukaran informasi yang diatur dengan baik, maka informasi mengenai kondisi suatu jaringan dapat diambil dan kemudian digunakan untuk dianalisis. Informasi ini sangat berguna bagi para *network administrator* untuk melakukan pengaturan performa jaringan, melakukan *troubleshooting* jika ada masalah, atau bahkan dapat digunakan untuk merencanakan perkembangan jaringannya.

Protokol SNMP ini kali pertama di-*develop* dan dikeluarkan pada tahun 1998 dan langsung menjadi sebuah solusi yang diminati banyak orang. Protokol ini sebenarnya tidak memiliki standar secara tertulis, namun dengan sendirinya menjadi standar *de facto* karena banyak digunakan pada dunia nyata. SNMP merupakan solusi yang sederhana, tidak membutuhkan *coding* yang rumit untuk diimplementasikan, menjadikannya banyak digunakan oleh vendor-vendor perangkat jaringan untuk dipasang di perangkat buatan mereka.

Selain itu, SNMP juga mudah dikembangkan dan fleksibel dibangun di mana-mana karena tidak tergantung pada arsitektur *hardware*. Hal inilah menyebabkan vendor-vendor dapat dengan mudah membangun sebuah SNMP agent di dalam produk mereka, sehingga dapat dijadikan sebagai pendongkrak nilai jual. Dan yang lebih hebat lagi, sistem monitoring menggunakan SNMP tidak hanya dapat dibangun di perangkat jaringan dan perangkat komputer seperti printer, modem, server, dan banyak lagi, melainkan pada perangkat-perangkat elektronik dan rumah tangga seperti UPS, AC, sistem PABX, dan banyak lagi. Mungkin saat ini kebanyakan perangkat yang menggunakan IP memiliki SNMP *Agent* di dalamnya.

Apa saja yang dapat dimonitor SNMP? Cukup banyak. Mulai dari *transfer rate* dari suatu *interface*, *CPU load*, *Error-error* yang ada dalam suatu perangkat, temperatur sebuah perangkat, dan banyak lagi. Selama SNMP Agent ada di dalam sebuah perangkat dan parameter-parameter yang ingin di monitor di sana disertakan, maka Anda dapat melihatnya dengan jelas.

Protokol SNMP sudah melalui beberapa versi pembuatan, di antaranya adalah:

- **SNMP version 1 (SNMPv1)**

Protokol SNMPv1 ini adalah protokol SNMP yang sudah menjadi standar dan banyak digunakan saat ini. SNMPv1 ini mengandalkan atribut yang disebut *Community* untuk menjaga keamanannya. Atribut Community ini adalah berupa sebuah *plain-text* karakter yang fungsinya tidak lain adalah sebagai sebuah *password*. Aplikasi SNMP apapun yang mengetahui atribut Community dari suatu jaringan bisa mendapatkan akses ke dalam database informasi dari jaringan tersebut. Ada tiga jenis Community dalam SNMPv1 ini, yaitu *read-only*, *write-only*, dan *trap*.

- **SNMP version 2 (SNMPv2c)**

Protokol SNMP versi ini sering disebut dengan *Community String-based SNMPv2*, di mana protokol versi ini mengombinasikan fitur baru dari SNMPv2 dengan sistem keamanan yang ada pada SNMPv1. Versi ini masih belum ada standar baku dari IETF dan masih bersifat eksperimental. Meskipun demikian, sudah ada beberapa vendor yang menggunakan SNMP versi ini untuk perangkatnya.

- **SNMP version 3 (SNMPv3)**

SNMP protokol versi ini diklaim akan menjadi versi selanjutnya yang banyak digunakan. Namun, sampai sekarang

masih belum distandardisasi secara penuh dan belum dipublikasikan oleh IETF. Dalam versi ini, protokol SNMP ditambahkan kemampuan untuk mendukung otentikasi yang kuat dan komunikasi privat antara komponen-komponen jaringan monitoring.

Arsitektur Network Management dengan SNMP

Jaringan yang dilengkapi dengan sistem *Network Management* dan monitoring yang menggunakan SNMP terdiri dari tiga komponen kunci, yaitu *Managed device*, *Agents*, dan *Network Management System* (NMS).

Managed Device

Adalah sebuah perangkat atau titik jaringan yang di dalamnya terdapat kemampuan *Agent* dan berlokasi di dalam sebuah jaringan yang di-*manage*. *Managed device* bertugas untuk mengumpulkan data berupa informasi manajemen dan menyetorkannya ke sebuah NMS dengan menggunakan protokol SNMP. Managed device kebanyakan adalah berupa perangkat jaringan seperti router, switch, hub, server, dan banyak lagi perangkat lainnya.

Agent

Adalah sebuah modul *software management* jaringan yang terdapat di dalam sebuah managed device. Sebuah perangkat yang disertai dengan agent yang memiliki kemampuan mengumpulkan informasi lokal dari dirinya sendiri dan kemudian mengubah bentuknya menjadi kompatibel dengan SNMP. Software monitoring ini dapat berupa sebuah program terpisah, seperti SNMP daemon pada sistem berbasiskan UNIX, atau merupakan fasilitas yang sudah terintegrasi misalnya seperti pada IOS produk Cisco, *low-level* OS pada UPS, dan banyak lagi. Ciri-ciri perangkat yang bertindak sebagai Agent adalah:

- Mengimplementasikan seluruh protokol SNMP.
- Mengumpulkan dan menyetor data yang terdapat pada *Management Information Base*.
- Dapat membangun komunikasi secara asinkron ke NMS untuk mengirimkan sinyal suatu kejadian.

Network Management System

Adalah sebuah perangkat yang bertindak sebagai *manager* dari *Agent* yang mengeksekusi aplikasi untuk monitoring dan *control*. Semua informasi yang dibawa dengan SNMP dari sebuah *Managed device* akan diambil oleh perangkat ini dan kemudian diolah lebih lanjut untuk diubah menjadi informasi yang berguna bagi *network administrator*.

Perangkat NMS harus memiliki kemampuan *processing* dan memory yang besar. Satu atau lebih NMS harus ada pada setiap jaringan yang ingin dimonitoring. Beberapa vendor yang membuat *platform* khusus untuk *network management* adalah Hewlett Packard OpenView, Dec PolyCenter Network Manager, IBM AIX NetView/6000, SunConnect SunNet Manager, dan banyak lagi. Ciri-ciri perangkat yang bertindak sebagai NMS adalah:

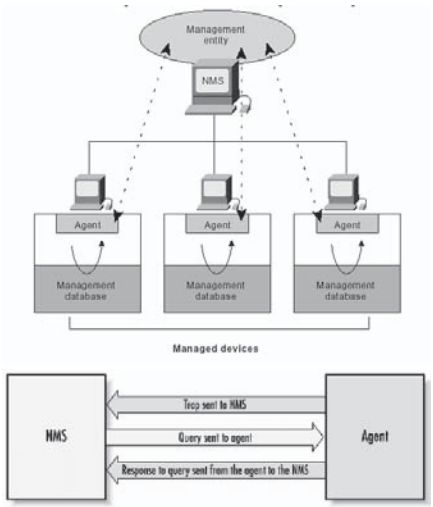
- Menjalankan aplikasi monitoring dan melakukan proses *poll* dan *trap*.
- Mengimplementasikan seluruh protokol SNMP.
- Memiliki kemampuan mencari informasi dari *Agent* (*Query* atau *poll*).
- Menerima respon dari Agent (Trap).
- Menentukan variabel pada Agent (Set).
- Melakukan *Acknowledgement* terhadap signal kejadian yang dikirim oleh Agent.

Bagaimana Protokol SNMP Mengumpulkan Data?

Proses pengumpulan data dalam protokol SNMP terdiri dari beberapa langkah yang cukup panjang. Langkah-langkah ini sering disebut sebagai *Protocol Data Unit* (PDU) karena masing-masing langkah memiliki format pesan dan cara kerjanya sendiri-sendiri. Berikut ini adalah proses pengumpulan datanya (proses tambahan pada protokol SNMPv2 dan SNMP v3 tidak bahas):

Operasi GET

Operasi GET biasanya dilakukan oleh sebuah perangkat NMS, di mana perangkat tersebut mengirimkan sebuah *request* ke perangkat yang bertindak sebagai *Agent*. Dalam proses ini, NMS akan meminta data dari sebuah objek yang ada dalam agent tersebut. Kemudian perangkat Agent akan menerimanya sebisa mungkin. Jika request sampai pada perangkat yang sedang dalam beban pekerjaan berat, seperti misalnya router, maka perangkat tersebut tidak akan dapat membalas operasi ini. Request ini akan dijawab dengan PDU GET-Response dan dikirim kembali ke NMS apabila telah melalui beberapa proses di perangkat Agent tersebut.



Jaringan yang siap untuk dimonitoring biasanya terdapat ketiga komponen-komponen kunci ini di dalamnya.

Operasi GET Next

Operasi ini digunakan ketika perangkat NMS ingin meminta beberapa data objek dari sebuah perangkat. Dengan beberapa PDU GET Next, sebuah NMS dapat meminta seluruh objek yang ada di sebuah *Agent*. Biasanya PDU ini harus didahului dulu dengan operasi GET, baru kemudian diikuti dengan GET Next.

Operasi SET

PDU SET memiliki fungsi untuk melayani NMS dalam melakukan perubahan nilai sebuah objek pada agent atau membuat baris data baru pada tebel di sebuah *Agent*.

Operasi TRAP

Operasi TRAP biasanya dilakukan oleh sebuah *Agent* untuk menginformasikan suatu kejadian yang dialaminya pada NMS. Dalam proses TRAP ini, NMS tidak mengirimkan *acknowledge* untuk data yang dikirimkan, sehingga Agent tidak akan tahu informasi yang dikirimnya sampai atau tidak ke tujuannya.

Bagaimana Data Tersebut Setelah Diambil?

Untuk mengolah informasi yang diterima dari setiap perangkat melalui protokol SNMP, data tersebut harus dapat diakses secara logika. Dapat diakses secara logika maksudnya adalah, informasi-informasi tersebut harus disimpan di suatu tempat untuk dapat diambil, kemudian diproses dan dimodifikasi secara logika oleh sistem.

Untuk memungkinkan hal itu, ada dua komponen penting yang mengatur agar data yang diterima, dimengerti, dan dapat diolah lebih lanjut. Dua komponen tersebut adalah:

1. Structure of Management Information (SMI)

Structure of Management Information adalah sebuah sistem yang mengatur pendefinisian dari objek-objek yang dimonitor beserta sifat-sifatnya. Setiap Agent pasti memiliki sekumpulan daftar dari objek-objek yang di-*manage* dan dimonitor olehnya. Salah satu contoh objek adalah status operasional dari *interface* sebuah router, misalnya status interface sedang *UP, Down*, atau *Testing*. SMI mengatur penamaan dan deskripsi informasi dari objek-objek yang ada, sehingga proses logika bisa berjalan. SMI mengharuskan semua objek memiliki sebuah nama (Object Identifier, OID), sistem *Syntax* dan sistem *Encoding* masing-masing.

- **Apakah OID?**

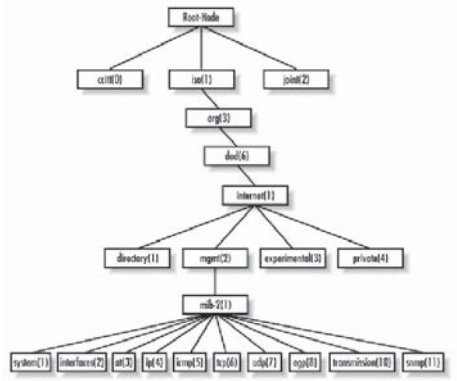
Nama dari objek-objek ini sering disebut sebagai *Object Identifier* (OID) yang bersifat unik pada masing-masing objek. Penamaan ini terbagi dalam dua bentuk, penamaan numerik dan penamaan '*human readable*' yang dapat langsung dibaca oleh manusia. Kedua penamaan ini sangat panjang dan tidak nyaman dibaca, namun aplikasi-aplikasi SNMP dapat membuatnya lebih mudah dan enak untuk dimengerti.

Skema penamaan dari objek-objek ini berbentuk hirarki seperti akar pohon. Masing-masing penamaan tersebut kemudian di beri nomor-nomor yang bertugas untuk mewakili nama objek tersebut. Semakin ke bawah, maka akan semakin banyak nomor yang dilewati. Kemudian urutan nomor-nomor inilah yang dijadikan sebagai Objek ID dari sebuah objek. Maka dari itu, wujud dari objek ID adalah sebaris nomor-nomor yang dipisahkan oleh tanda titik (.). Meskipun OID dapat dibaca oleh manusia, namun akan cukup sulit untuk diartikan, apalagi oleh orang awam hanya dianggap sebagai sekumpulan nomor saja.

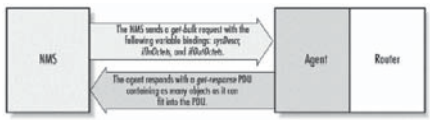
Di dalam sebuah pohon objek, sebuah titik yang paling atas biasa dinamai sebagai *root* dan cabang-cabangnya disebut sebagai *subtree* dan sebuah titik tanpa cabang sama sekali dinamai *leaf*. Seperti terlihat pada gambar, root memiliki tiga buah subtree, yaitu CCITT(0), ISO(1), dan Joint(2). Namun subtree yang berhubungan dengan SNMP hanyalah subtree ISO, maka itu kedua subtree yang lain tidak di bahas di sini. Karena ISO memiliki penomoran 1, maka untuk selanjutnya sebuah objek ID akan dimulai dari angka 1 yang menandakan subtree ini. Dan selanjutnya penggunaan SNMP untuk monitoring jaringan sehari-hari akan banyak berkutat pada subtree iso(1).org(3).dod(6).internet(1). Maka dari itu, kebanyakan OID yang digunakan sehari-hari adalah berawalan angka 1.3.6.1.

Selanjutnya setelah subtree internet(1), objek yang berada di bawahnya adalah dictionary(1), management(2), experimental(3), dan private(4). Subtree-subtree inilah yang cabang-cabangnya nantinya banyak dipakai dalam melakukan monitoring. Setelah sampai pada objek tertentu yang dituju, maka akan didapatkan sebuah deskripsi objek tersebut yang biasanya berisikan data yang berhubungan dengan objek itu sendiri. Kemudian data inilah yang dipakai untuk ditampilkan ke sang administrator.

Salah satu cabang dari objek private(4) yang diberikan khusus untuk menamai objek-objek khusus dari para vendor perangkat jaringan adalah subtree enterprises(1). Jadi masing-masing vendor memiliki penamaan sendiri dan diletakan di subtree ini. Misalnya untuk produk-produk jaringan buatan Cisco, OID yang diberikan untuk Cisco adalah Cisco(9). Jadi untuk menandai objek-objek yang khusus



Pohon objek inilah yang digunakan untuk menghasilkan sebuah Objek ID yang berupa deretan angka-angka yang dipisahkan oleh titik.



Operasi GET dari perangkat NMS akan dijawab dengan PDU GET-Response yang biasanya berisikan data-data sebuah objek.

berada di produk Cisco, OID nya adalah 1.3.6.1.4.1.9. Objek private enterprises ini diatur dan dikendalikan oleh IANA.

- **Apakah Sistem Syntax?**

Syntax mendefinisikan tipe data dari sebuah oktet data pada objek, seperti misalnya *integer* atau *string*. Syntax-syntax ini merupakan sebuah bahasa komunikasi antara *Agent* dengan *manager* (NMS). Sistem syntax yang digunakan untuk SNMP adalah *Abstrac Syntax Notation One* (ASN.1) yang bersifat independen, artinya semua *platform* dapat mengerti syntax tersebut (misalnya syntax pada mesin Windows dapad dimengerti oleh mesin berbasis Sun).

- **Apakah Sistem Encoding?**

Encoding menjelaskan bagaimana informasi yang berasal dari sebuah perangkat di-*encode* dan di-*decode* untuk kemudian ditransmisikan antar-mesin melalui media seperti Ethernet, misalnya. Sistem encoding yang digunakan pada SNMP adalah *Basic Encoding Rules* (BER). Yang akan banyak dibahas selanjutnya adalah bagaimana OID berguna bagi proses manajemen dan monitoring.

2. Management Information Base (MIB)

Management Information Base dapat

Tabel 1.

Subtree	OID	Deskripsi
system	1.3.6.1.2.1.1	Berisikan objek-objek yang berhubungan dengan informasi sistem, misalnya system uptime, system contact, system date, dan sebagainya.
interface	1.3.6.1.2.1.2	Berisikan objek yang menggambarkan keadaan interface-interface yang ada pada sebuah perangkat, misalnya status up atau down, byte sent dan byte receives, dan banyak lagi.
at	1.3.6.1.2.1.3	Subtree ini berisikan informasi mengenai address translation group.
ip	1.3.6.1.2.1.4	Semua aspek informasi mengenai IP ada di dalam subtree ini, seperti misalnya IP routing.
icmp	1.3.6.1.2.1.5	Informasi lalu-lintas dan proses yang berhubungan dengan paket icmp dapat ditemukan di subtree ini.
tcp	1.3.6.1.2.1.6	Memantau semua aspek yang berhubungan dengan koneksi TCP, seperti TCP SyncSent, Closed, Listen, dan sebagainya.
udp	1.3.6.1.2.1.7	Memantau statistik dari proses komunikasi UDP.
egp	1.3.6.1.2.1.8	Mengumpulkan informasi mengenai statistik EGP dan juga menyimpan EGP neighbour table.
transmission	1.3.6.1.2.1.10	Sebenarnya tidak banyak digunakan pada group ini, namun akan banyak digunakan pada MIB yang khusus berisikan media yang spesifik.
snmp	1.3.6.1.2.1.11	Memantau implementasi dari SNMP pada sebuah perangkat, seperti paket SNMP yang keluar-masuk dan sebagainya.

dideskripsikan sebagai database dari objek-objek yang dikumpulkan oleh *Agent*. Semua status atau data statistik yang dapat diakses oleh NMS disebutkan dalam MIB. Kalau SMI menyediakan cara untuk mendeskripsikan suatu objek, MIB merupakan kumpulan dari definisi objek-objek tersebut. Bagaikan sebuah kamus, SMI bagaikan cara-cara mengeja sebuah kata beserta arti-artinya dalam penggunaan, sedangkan MIB adalah sebuah kata yang dijelaskan itu sendiri. Setiap *Managed device* menyimpan database dari objek-objek yang dideskripsikan oleh SMI dalam sebuah MIB.

Setiap perangkat dapat menyertakan banyak MIB di dalamnya, dan mungkin saja masing-masingnya memiliki MIB yang berbeda satu sama lain. Misalnya produk router Cisco seri 1700 memiliki MIB yang berbeda dengan seri 2600. Keduanya memiliki karakteristik masing-masing untuk dijelaskan di dalam MIB nya. Namun, setiap perangkat pasti menerapkan apa yang disebut MIB-II. MIB-II merupakan MIB standar yang di dalamnya berisikan objek-objek standar seperti statistik *interface* dan banyak lagi. Tujuan utama dari MIB-II ini adalah untuk menciptakan informasi TCP/IP yang general pada semua perangkat tanpa menutup kemungkinan bagi semua perangkat tersebut untuk meyertakan MIB lain di dalamnya.

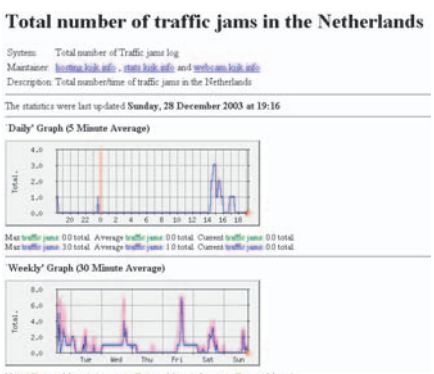
MIB-II terletak di bawah subtree Management. Jika dituliskan dalam OID, maka subtree MIB-II dapat dijangkau dengan menggunakan OID 1.3.6.1.2.1.2. Di bawah subtree MIB-II terdapat cukup banyak objek yang dapat Anda gunakan untuk memantau jalannya suatu perangkat jaringan. Objek-objek tersebut dapat dilihat pada Tabel 1.

Setelah semua data yang diambil dari Agent diterima oleh NMS, maka data tersebut kemudian akan diproses lebih lanjut dengan menggunakan aplikasi-aplikasi *trend analys* yang dapat sekaligus menampilkan grafik yang dapat dilihat dengan lebih mudah oleh penggunanya. Aplikasi seperti *Multi Router Traffic Grapher* (MRTG) merupakan aplikasi yang cukup banyak digunakan oleh para administrator untuk menangkap dan menggambarkan proses yang sedang terjadi pada sebuah jaringan.

Monitoring Sangat Penting

Menjaga dan merawat jaringan dengan melakukan monitoring sangatlah penting untuk dilakukan oleh setiap *network administrator*. Sebuah jaringan yang dapat dipercaya untuk melayani kebutuhan para penggunanya adalah jaringan yang sehat dan bekerja optimal, tidak hanya sekadar berjalan saja. Dengan mempelajari *trend* dan *behaviour*-nya dengan menggunakan protokol SNMP, sebenarnya Anda sudah berada di jalur yang tepat untuk membuat sebuah jaringan yang *predictable*. Anda dapat mengetahui lebih dulu kekurangan, kelebihan, dan juga permasalahan yang ada di dalamnya, meski memang tidak semuanya dapat di pantau.

Hanya saja, teori dan penerapannya tidak sesederhana namanya *Simple Network Management Protocol*. Anda harus mempelajarinya dengan lebih teliti untuk dapat menerapkannya. Selamat belajar! ■



Dengan hanya membuka sebuah halaman web, Anda dapat memantau apa yang sedang terjadi pada jaringan Anda. Tentu ini akan sangat membantu Anda dalam melakukan tindakan ke depan.

Administrasi Jaringan Linux

Gunung Sarjono

Beberapa orang menganggap bahwa jaringan Linux adalah suatu hal yang sulit. Namun, hal tersebut tidaklah sepenuhnya benar. Melalui artikel ini, Anda dapat melihat hal-hal yang berhubungan dengan pengaturan jaringan Linux.

Alamat IP digunakan untuk mengidentifikasi *interface* jaringan pada komputer. Alamat ini tidak diberikan secara sembarangan. Setiap alamat IP mengandung dua informasi, yaitu alamat jaringan dan alamat *interface* komputer. Sebagai contoh, 192.168.1.4 merupakan alamat IP pada jaringan 192.168.1.4 dan alamat interface-nya, yang kadang disebut alamat host adalah .4. Dengan kata lain, alamat IP ini merujuk ke interface jaringan .4 pada jaringan 192.168.1.0. Bagian mana yang mengidentifikasi jaringan bergantung kepada kelas jaringan tersebut? Alamat IP berbagai kelas jaringan dapat Anda lihat pada tabel.

Bagaimana Mengonfigurasi Interface Jaringan?

Pada bagian ini, kami asumsikan Anda telah memiliki kartu Ethernet yang telah terinstalasi. Untuk mengetahui daftar NIC yang kompatibel dengan Red Hat Anda, lihat bagian kompatibilitas *hardware* pada situs Red Hat (*http://hardware.redhat.com/hcl*).

Sebelum memulai, jalankan program dmesg (atau lihat file /var/log/dmesg) untuk melihat apakah NIC Anda dikenali oleh kernel Linux. Jika ya, pada *output* dmesg Anda akan melihat beberapa baris mengenai perangkat jaringan Anda dan Anda siap untuk mengonfigurasinya.

Pada contoh ini, kami asumsikan Anda ingin membuat konfigurasi yang *persistent*, yang akan digunakan setiap kali menjalankan komputer. Hal ini dilakukan dengan bangunan *script* /etc/rc.d/init.d/network. Ketika script ini di-*link* (nama link-nya adalah S*xx*network; di mana *xx* adalah sembarang nomor) dari direktori run level default (dapat dari /etc/rc.d/rc3.d), script /etc/rc.d/init.d/network akan dijalankan sewaktu *boot*. Script ini memuat interface jaringan dengan file yang tersimpan pada direktori /etc/sysconfig.

Jadi, file pertama yang harus Anda modifikasi adalah /etc/sysconfig/network. Setelah mengonfigurasi file tersebut, Anda dapat membuat file interface jaringan pada direktori /etc/sysconfig/network-scripts. File ini menggunakan konvensi nama ifcfg-*interface*. Sebagai contoh, nama untuk interface jaringan eth0 adalah ifcfg-eth0. Sebelum mengonfigurasi file ini, Anda harus mengetahui alamat jaringan, *network mask*, dan alamat IP network interface host, dan alamat broadcast. Setelah mengonfigurasi file ini, Anda dapat mengaktifkan interface tersebut. Cara paling mudah untuk melakukannya adalah dengan menjalankan perintah /etc/sysconfig/network-scripts/ifup eth0.

Script ifup tersebut mengambil nama perangkat sebagai argumen dan menjalankannya. Ia juga membuat rute *default* untuk jaringan tersebut. Setelah menjalankan perintah ini, Anda dapat menggunakan program ifconfig untuk melihat apakah interface sudah aktif dan berjalan. Anda dapat menggunakan ifconfig tanpa argumen apapun untuk melihat daftar lengkap perangkat jaringan yang aktif.

Ingat bahwa script ifup menggunakan ifconfig untuk mengaktifkan interface.

Konfigurasi File /etc/sysconfg/network

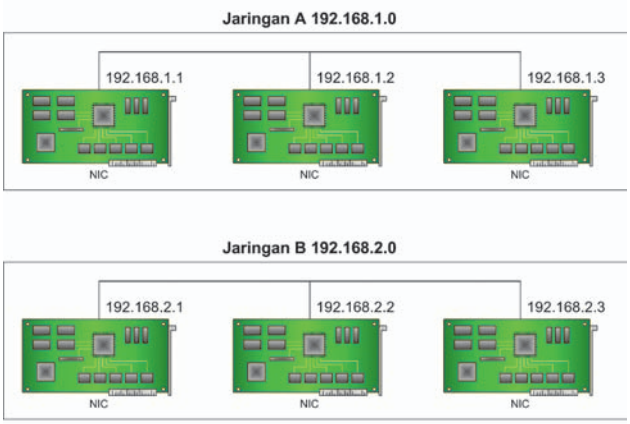
```
NETWORKING=yes
FORWARD_IPV4=yes
HOSTNAME=genesis.pcmedia.co.id
DOMAINNAME=pcmedia.co.id
GATEWAY=192.168.1.1
GATEWAYDEV=eth0
```

Berikut adalah penjelasan entri di atas:

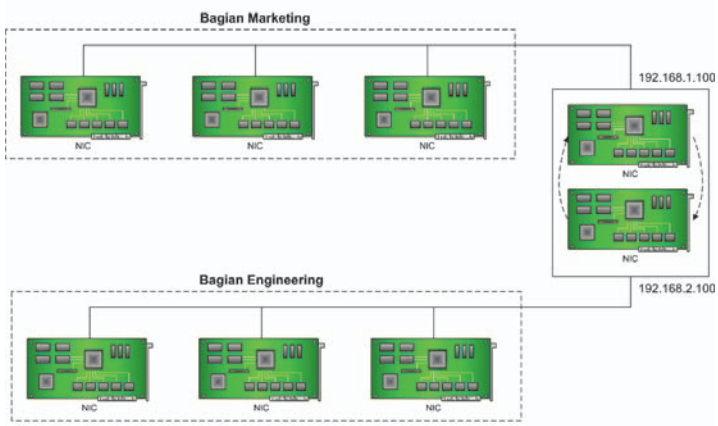
- Baris pertama menyatakan bahwa Anda ingin meng-*enable* dukungan jaringan.
- Baris kedua menyatakan penerusan paket IP. Baris ini harus diset ke *yes* jika Anda ingin menjalankan penerusan paket IP ke dan dari server, dan hanya diperlukan jika server berperan sebagai *gateway* atau router.
- Dua baris berikutnya menyatakan nama host dan nama domain jaringan Anda. Nama host harus termasuk dalam nama domain. Dengan kata lain, jangan gunakan nama host dengan nama domain yang berbeda dari DOMAINNAME.
- Dua baris terakhir menyatakan informasi yang diperlukan tentang default gateway. Default gateway merupakan komputer yang mentransfer paket ke dan dari komputer Anda. GATEWAYDEV penting ketika Anda menggunakan beberapa *interface* jaringan. Jika Anda hanya menggunakan satu interface, baris ini selalu diset ke eth0. Jika Anda menggunakan beberapa interface, set baris ini ke nama interface (eth0, eth1, dan seterusnya) yang terhubung ke *default gateway*.

Range Alamat IP.

Kelas	Range Alamat	Subnet Mask
A	0.0.0.0 - 127.255.255.255	255.0.0.0
B	128.0.0.0 - 191.255.255.255	255.255.0.0
C	192.0.0.0 - 255.255.255.255	255.255.255.0
D	224.0.0.0 - 239.255.255.255	-
E	240.0.0.0 - 245.255.255.255	-



Dua jaringan yang berbeda.



Dua jaringan yang berbeda digabungkan dengan *gateway*.

Sebagai contoh, ifconfig eth0 192.168.1.4 netmask 255.255.255.0 broadcast 192.168.1.255 akan mengaktifkan eth0 dengan IP 192.168.1.4. Anda juga dapat menonaktifkan interface dengan menggunakan perintah ifconfig. Sebagai contoh, perintah ifconfig eth0 down akan menonaktifkan interface eth0.

Setelah melihat interface aktif dan berjalan, coba hubungi komputer pada jaringan dengan menggunakan perintah ping. Jika saat instalasi Anda belum men-*set up* file /etc/resolv.conf, Anda tidak dapat mem-ping host dengan menggunakan nama host-nya. File ini di-*setup* ketika instalasi Linux Red Hat. Jika Anda memberikan alamat IP *name server* yang tidak valid, komputer Anda tidak akan dapat menghubungi name server tersebut.

Namun, jika tidak dapat mem-ping host dengan menggunakan alamat IP atau host name, Anda mungkin mengalami masalah *routing*. Jika mengaktifkan interface dengan menggunakan ifconfig, bukan script ifup, Anda perlu membuat rute default ke jaringan dan *gateway* Anda secara manual.

Untuk membuat rute default jaringan Anda, gunakan perintah route sebagai berikut route add -net *network address netmask device*. Sebagai contoh, untuk membuat rute ke jaringan 192.168.1.4 dengan netmask 255.255.255.0 dan interface eth 0, jalankan route add -net 192.168.1.4 255.255.255.0 eth0.

Untuk mengeset gateway default, jalankan perintah route sebagai berikut route add default gw *gateway address device*. Sebagai contoh, untuk mengeset gateway default ke 192.168.1.1, jalankan perintah route add default gw 192.168.1.1 eth0.

Untuk memeriksa apakah rute jaringan dan gateway default Anda, sudah terdapat pada tabel routing, jalankan perintah route -n. Sampai di sini, gunakan program ping untuk mem-ping host yang berada di dalam dan di luar jaringan Anda.

Apakah Gateway Default?

Pada gambar tampak dua jaringan yang benar-benar terpisah. Komputer pada tiap jaringan dapat melihat *host* pada jaringan yang sama. Seorang *user* dengan menggunakan komputer 192.168.1 pada jaringan A dapat mem-*ping* kedua komputer lainnya. Namun, user tersebut tidak dapat mengakses host manapun pada jaringan B, karena tidak ada jalan bagi paket dari jaringan A ke jaringan B. Karena dua jaringan yang berbeda secara fisik dapat dihubungkan bersama, kedua jaringan ini dapat digabungkan dengan menggunakan *gateway*.

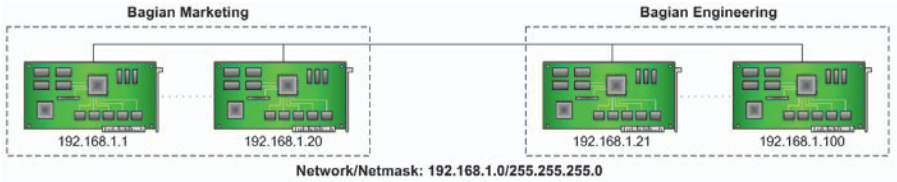
Komputer gateway memiliki dua kartu jaringan. Kartu jaringan yang terhubung ke jaringan A memiliki alamat IP 192.168.1.100, dan yang terhubung ke jaringan B memiliki alamat IP 192.168.2.100. Sebagai contoh, ketika komputer 192.168.1.1 ingin mengirim paket ke 192.168.2.1, ia meneruskan paket tersebut ke gateway 192.168.1.100. Gateway ini

Konfigurasi File /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.1.127
IPADDR=192.168.1.126
NETMASK=255.255.255.128
NETWORK=192.168.1.0
ONBOOT=yes
```

Berikut adalah penjelasan entri di atas:

- Baris pertama menyatakan nama perangkat.
- Baris kedua menyatakan option BOOTPROTO. Pilih *static* jika Anda ingin memberikan alamat IP secara manual (statis).
- Baris ketiga menyatakan alamat *broadcast*.
- Baris keempat menyatakan alamat IP komputer.
- Baris kelima menyatakan alamat *network mask* (netmask).
- Baris keenam menyatakan alamat jaringan.
- Baris ketujuh menyatakan option ONBOOT. Pilih *yes* jika Anda ingin interface jaringan "up" (diaktifkan) setelah boot.



Jaringan yang terdiri dari 100 *host*.

kemudian meneruskan paket tersebut (yang telah dimodifikasi) ke interface lainnya (192.168.2.100), yang kemudian akan mengirimnya ke komputer tujuan. Ketika komputer tujuan ingin memberi respon, ia mengikuti proses yang sama, sehingga gateway dapat mengirim paket ke komputer asal pada jaringan A.

Jadi, gateway hanya dibutuhkan jika Anda ingin berkomunikasi dengan jaringan yang berbeda. Gateway tidak harus sebuah PC atau *workstation*, tetapi dapat juga berupa *hardware* khusus, seperti router.

Apakah Subnet?

Jaringan yang tampak pada gambar memiliki 100 host. Karena semua komputer berbagi pakai kabel jaringan yang sama, maka sudah dapat diperkirakan *traffic*-nya akan tinggi pada jam kerja. Sebagai contoh, jika semua orang pada bagian *marketing* dan bagian *engineering* menggunakan komputer secara bersamaan, segalanya akan menjadi lambat, karena semua host berbagi pakai *bandwidth* jaringan. Solusi untuk mengatasi hal ini adalah dengan membagi jaringan ke dalam subnet.

Jaringan 192.168.1.0 telah dibagi ke dalam dua subnet— 192.168.1.0 dan 192.168.1.128, yang dihubungkan dengan menggunakan *gateway* (atau router). Dengan cara ini, traffic antarkedua jaringan tersebut akan terpisah. Sebagai contoh, ketika user pada bagian marketing ingin berkomunikasi dengan komputer lain pada bagian tersebut, traffic antarkedua komputer ini tetap berada pada jaringan mereka. Hal yang sama juga berlaku pada bagian engineering. Komunikasi komputer antarbagian juga dapat dilakukan. Traffic internetwork ini diteruskan dari dan ke jaringan oleh gateway. Hasilnya adalah jaringan yang lebih cepat, karena semua user tidak lagi berada pada satu jaringan besar yang sama.

Ketika jaringan 192.168.1.0, dibagi ke dalam subnet, maka komputer harus menentukan cara bagaimana mengetahui alamat IP jaringan. Sebagai contoh, komputer dengan alamat IP 192.168.1.1 pada bagian marketing ingin mengetahui apakah 192.168.1.129 berada pada jaringan yang sama sehingga ia dapat menentukan apakah ia harus mengirim paket ke gateway atau tidak. Ia dapat mengetahui alamat IP milik subnet-nya melalui *network mask* atau *netmask*.

Sebelum membuat subnet, Anda perlu mengetahui bagaimana membuat netmask. Sebagai contoh, misalkan Anda ingin membuat network mask untuk jaringan 192.168.1.0. Alamat ini jika diubah ke dalam biner menjadi 11000000.10101000.00000001.00000000.

Karena ini merupakan alamat kelas C, maka bit jaringannya adalah 24-bit (3 byte) pertama. Untuk membuat mask, Anda harus mengeset semua bit tersebut ke 1 dan mengeset bit sisanya (host) ke 0. Jadi, netmask-nya adalah 11111111.11111111.11111111.00000000 yang dalam desimal sama dengan 255.255.255.0.

Jadi, ketika membuat netmask, Anda cukup mengingat bahwa bit jaringan semuanya 1 dan bit host semuanya 0. Sebagai contoh, netmask untuk jaringan 10.0.0.0 (kelas A) adalah 255.255.255.0, sedangkan netmask untuk jaringan 129.1.0.0 (kelas B) adalah 255.255.0.0.

Pada jaringan TCP/IP, ketika komputer ingin mengirimkan suatu paket, ia akan menentukan apakah alamat yang dituju merupakan alamat lokal atau bukan. Hal ini dilakukan dengan operasi AND. Sebagai contoh, untuk menentukan apakah 192.168.1.1 merupakan alamat lokal pada jaringan dengan subnet mask 255.255.255.0, komputer host akan melakukan operasi AND antara 11000000.10101000.00000001.00000001 (192.168.1.1) dengan 11111111.11111111.11111111.00000000 (255. 255.255.0) sehingga didapat 192. 168.1.0.

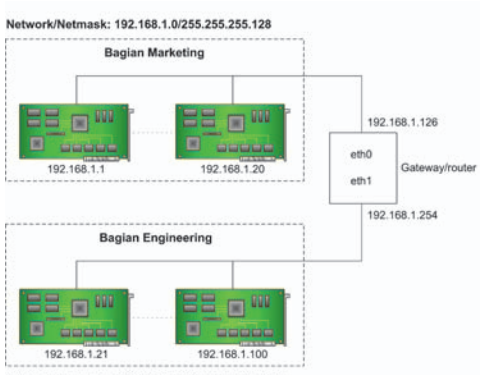
Seperti yang Anda lihat, operasi AND alamat IP 192.168.1.1 menghasilkan alamat jaringan 192.168.1.0. Karena komputer host tahu pada jaringan mana ia berada, ia dapat langsung menentukan bagaimana menangani paket untuk alamat IP tersebut.

Ketika jaringan kelas C dibiarkan secara utuh, netmask-nya adalah 255.255. 255.0. Untuk membagi jaringan tersebut ke dalam subnet, Anda perlu membuat netmask baru dengan mengeset satu atau beberapa bit host pada netmask default (255.255.255.0) ke 1. Dengan mengeset *most significant bit* (MSB) pada contoh sebelumnya ke 1, Anda akan mendapatkan 11111111.11111111. 11111111. 10000000 (255.255.255.128).

Ini adalah netmask yang membagi jaringan menjadi dua. Semua alamat IP dari 192.168.1.0 sampai 192.168.1.127 berada pada jaringan 192.168.1.0, dan sisa alamat IP lainnya, 192.168.1.128 sampai 192.168.1.255, berada pada jaringan 192.168.1.128. Kedua jaringan menggunakan netmask 255.255.255.128.

Sebagai contoh, untuk menentukan keberadaan alamat IP 192.168.1.10, lakukan operasi AND antara 11000000. 10101000.00000001.00001010 (192. 168.1.10) dengan 11111111.11111111. 11111111.10000000 (255.255.255.128) sehingga didapat 192.168.1.0. Netmask 255.255.255.0 menunjukkan bahwa alamat IP 192.168.1.10 berada pada jaringan 192.168.1.0.

Sekarang, mari kita lihat di mana alamat IP 192.168.1.200 berada. Kita AND 11000000.10101000.00000001. 11001000 (192.168.1.200) dengan 11111111.11111111.11111111.10000000



Jaringan yang dibagi ke dalam subnet.

Konfigurasi File /etc/sysconfig/network-scripts/ifcfg-eth1 pada Gateway

```
EVICE=eth1
BOOTPROTO=static
BROADCAST=192.168.1.255
IPADDR=192.168.1.254
NETMASK=255.255.255.128
NETWORK=192.168.1.128
ONBOOT=yes
```

Konfigurasi File /etc/sysconfig/network-scripts/ifcfg-eth0 pada host

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.1.127
IPADDR=192.168.1.1
NETMASK=255.255.255.128
NETWORK=192.168.1.0
ONBOOT=yes
```

(255.255.255.128) sehingga didapat 192.168.1.128.

Netmask yang sama memberikan alamat jaringan yang berbeda (192.168.1.128) untuk alamat IP 192.168.1.200. Jadi, dengan menggunakan netmask ini, Anda telah membagi jaringan 192.168.1.0 ke dalam dua subnet - 192.168.1.0 dan 192.168.1.128.

Karena jaringan kelas C secara keseluruhan memiliki 256 alamat IP, maka setiap subnet tersebut memiliki 128 alamat IP. Namun, tidak semua alamat tersebut dapat digunakan. 192.168.1.0 (dengan netmask 255.255.255.0) digunakan sebagai alamat jaringan, dan 192.168.1.255 digunakan sebagai alamat *broadcast*. Alamat broadcast digunakan untuk berkomunikasi dengan semua host pada jaringan. Jadi, kedua alamat ini tidak dapat digunakan. Oleh karena itu, Anda hanya memiliki 254 alamat IP (256 - alamat IP jaringan - alamat IP broadcast).

Demikian juga pada jaringan yang terbagi ke dalam subnet, setiap subnet mendapatkan alamat jaringan dan alamat broadcast, jadi ketika jaringan kelas C dibagi menjadi dua, ada empat alamat IP yang tidak dapat digunakan, yaitu 192.168.1.0, 192.168.1.127, 192.168.1.128, dan 192.168.1.255.

Banyaknya subnet yang dapat Anda buat dapat diketahui dengan menggunakan rumus $2^{(bit\ host\ -\ 1)}$. Sebagai contoh, jumlah bit host pada kelas C adalah 8 (byte terakhir), jadi, secara teknis, ,Anda dapat membuat 2^7 (128) subnet. Namun, karena adanya beberapa alamat IP yang tidak dapat digunakan, banyak administrator jaringan yang tidak membuat lebih dari 16 subnet pada jaringan kelas C.

Jika Anda masih baru dengan konsep *subnetting*, tool ipcalc mungkin akan sangat berguna. Anda dapat menggunakan tool ini untuk memastikan perhitungan Anda sudah benar. Sebagai contoh, ketika menjalankan perintah ipcalc -n -b 192.168.1.129 255.255.255.128 kami mendapatkan BROADCAST =192.168.1.255 dan NETWORK=192.168.1.128.

Bagaimana dengan Konfigurasi Gateway?

Komputer *gateway* pada gambar subnet harus memiliki dua *interface*, yang terhubung ke setiap subnet. Konfigurasi file /etc/sysconfg/network dan file /etc/sysconfig/network-scripts/ifcfg-eth0 dapat dilihat pada bagian sebelumnya, sedangkan konfigurasi file /etc/sysconfig/network-scripts/ifcfg-eth1 dapat Anda lihat pada kotak keterangan.

Setting yang paling penting pada konfigurasi file /etc/sysconfig/network adalah FORWARD_IPV4=yes, yang mengaktifkan penerusan IP pada kernel. Anda dapat menggunakan perintah cat /proc/sys/net/ipv4/ip_forward untuk melihat apakah penerusan IP sudah dijalankan. Jika output-nya 1, berarti penerusan IP sudah diaktifkan; tetapi bila output-nya 0 berarti penerusan IP belum diaktifkan. Anda mengaktifkan atau menonaktifkan penerusan IP dengan memasukkan nilai 1 atau 0 dengan menggunakan perintah echo n > /proc/sys/net/ipv4/ip_forward, dimana n antara 1 atau 0.

Setelah file-file tersebut dikonfigurasi dan interface sudah berjalan (dengan *reboot* atau menggunakan perintah ifup), Anda perlu membuat dua rute default untuk kedua jaringan, dengan menggunakan perintah route add -net 192.168.1.0 255.255.255.128 eth0, dan route add -net 192.168.1.128 255.255.255.128 eth1. Perintah ini memastikan paket yang dikirim ke jaringan yang dimaksud akan dilewatkan melalui interface yang disebutkan.

Bagaimana dengan Konfigurasi Host?

Setiap *host* pada setiap subnet juga harus dikonfigurasi sehingga setiap komputer mengetahui alamat jaringan, subnet mask, dan juga *gateway*. Sebagai contoh, dengan asumsi setiap host adalah *workstation* Linux dan masing-masing memiliki satu Ethernet, maka file konfigurasi /etc/sysconfig/network setiap host pada jaringan 192.168.1.0 dapat Anda lihat pada kotak keterangan.

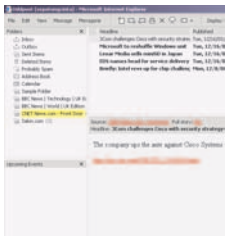
Dengan cara yang sama, host pada jaringan 192.168.1.128 juga perlu dikonfigurasi. Setiap host pada jaringan ini akan menggunakan GATEWAY=192. 168.1.254 pada file /etc/sysconfig/network-nya, dan menggunakan BROADCAST=192.168.1.255, dan NETWORK=192.168.1.128 pada file /etc/sysconfig/network-scripts/ifcfg-eth0-nya.

Tentu saja, jika ada host yang bukan Linux, seperti Windows 9x/NT, Anda harus menggunakan tool yang sesuai untuk mengeset konfigurasi tersebut. Setelah gateway dan host terkonfigurasi, Anda dapat mengetes jaringan. Cara paling mudah melakukannya adalah dengan menggunakan program ping. ■

Konfigurasi File /etc/sysconfig/network pada host

```
NETWORKING=yes
FORWARD_IPV4=yes
HOSTNAME=marketing-
1.pcmedia.co.id
DOMAINNAME=pcmedia.co.id
GATEWAY=192.168.1.126
```

Cara Baru Membaca Berita di Internet



Fadilla Mutiarawati

Berapa kali dalam sehari Anda membaca berita di Internet? Dan berapa portal berita yang Anda buka setiap harinya? Lebih dari satu, lima, atau bahkan lebih dari sepuluh? Jika memang sangat banyak, maka sudah saatnya Anda mulai menggunakan aplikasi atau fasilitas web yang bernama News Aggregator.

Membaca berita di Internet kadang dinilai lebih aktual dibandingkan melalui media lain. Hal ini disebabkan meng-*update* berita di Internet lebih mudah daripada media lain. Sehingga berita dalam Internet dinilai bergerak lebih cepat. Bandingkan saja isi koran Anda dengan situs berita seperti detik.com, mana yang lebih aktual? Hampir setiap menit, muncul berita baru yang dapat Anda baca lewat detik.com. Banyaknya variasi berita juga mengakibatkan tumbuhnya berbagai portal berita di internet. Mulai yang dengan CNet yang mengkhususkan berita teknologi sampai dengan E-Online yang memuat berita dari dunia *entertainment*. Sehingga tidak jarang banyak juga orang yang berlangganan lebih dari satu portal berita.

Untuk mempermudahkan seorang wartawan portal berita tersebut, maka diperkenalkan sebuah aplikasi yang bernama weblog. Sedangkan untuk memudahkan membacanya berita ada sebuah aplikasi yang dinamakan News Aggregator. Kedua-duanya merupakan salah satu aplikasi yang memanfaatkan teknologi RSS dari XML. Jika mengenai XML sudah pernah disinggung pada edisi sebelumnya, maka pada edisi kali ini akan dibahas mengenai News Aggregator, Weblog, dan RSS.

News Aggregator

Terlebih dahlu akan diinformasikan tentang News Aggregator, mengingat News Aggregator-lah yang paling dekat dengan masyarakat. Saat ini orang yang berlangganan portal maupun situs semakin banyak saja yang tidak jarang justru hanya untuk mencari berita terbaru saja, sehingga peranan News Aggregator sendiri semakin hari semakin dirasakan cukup penting dan sangat membantu.

Apa itu News Aggregator?

Adalah sebuah aplikasi yang akan menghimpun semua *headline* baru dari setiap portal berita yang menggunakan file RSS. News Aggregator banyak dimanfaatkan selain oleh perorangan juga oleh beberapa situs yang juga menyediakan layanan berita. Biasanya situs-situs ini menggunakan News Aggregator untuk memudahkan proses pencarian berita.

Aplikasi ini ada yang tersedia sebagai aplikasi *desktop* ada juga yang tersedia sebagai aplikasi *web base*.

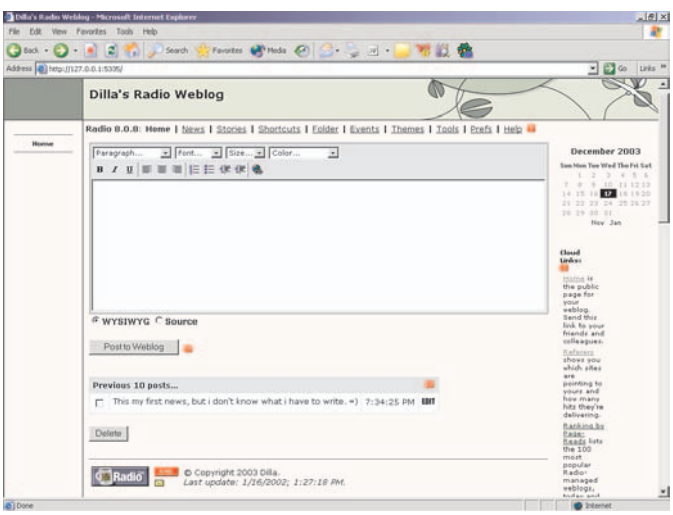
Mengapa menggunakan News Aggregator?

Biasanya untuk membaca sebuah berita seseorang akan mengunjungi satu situs berita ke satu situs lainnya. Bahkan saat ini ada juga sebagian masyarakat yang mengunjungi lebih dari 10 situs berita dalam sehari. Padahal sebagian berita biasanya bersumber dari tempat yang sama. Misalnya berita teknologi yang ada dalam YahooNews dengan GoogleNews keduanya bisa saja berasal dari CNet. Sedangkan untuk foto, keduanya dapat berasal dari sumber yang sama juga seperti AP atau Reuters.

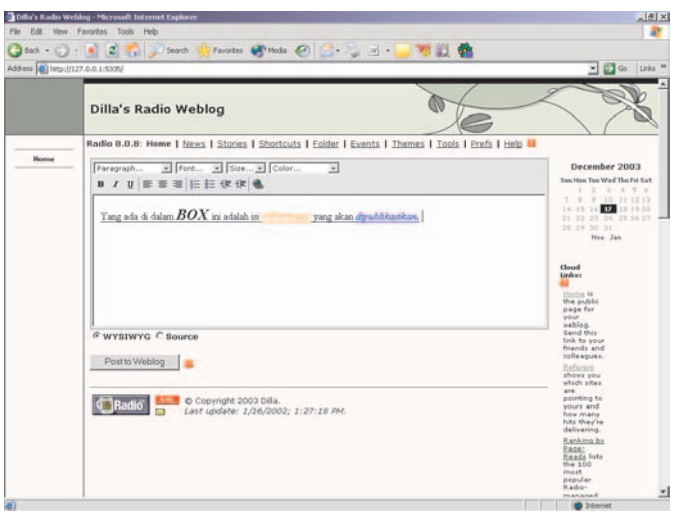
Sedangkan dengan adanya News Aggregator ini, Anda tidak perlu lagi mengunjungi setiap situs atau situs berita. Berita-berita yang akan baca akan datang dengan sendirinya kepada Anda melalui News Aggregator ini. Dengan menggunakan News Aggregator, selain dapat mempersingkat waktu, Anda juga akan menghemat tenaga.

Bagaimana cara kerja News Aggregator?

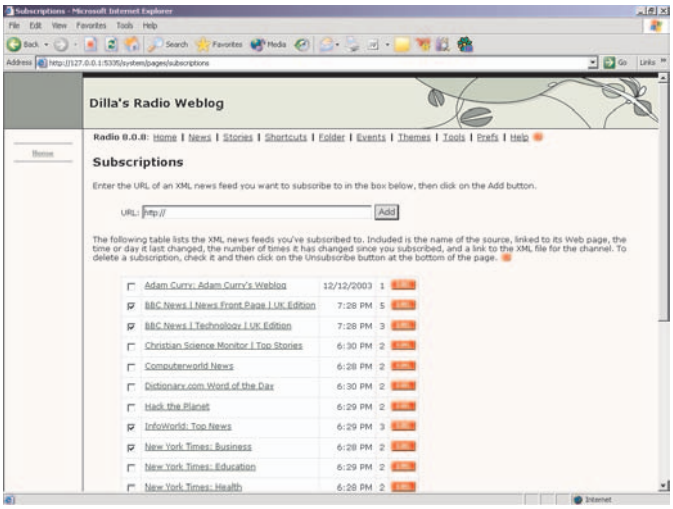
Setiap News Aggregator yang Anda miliki



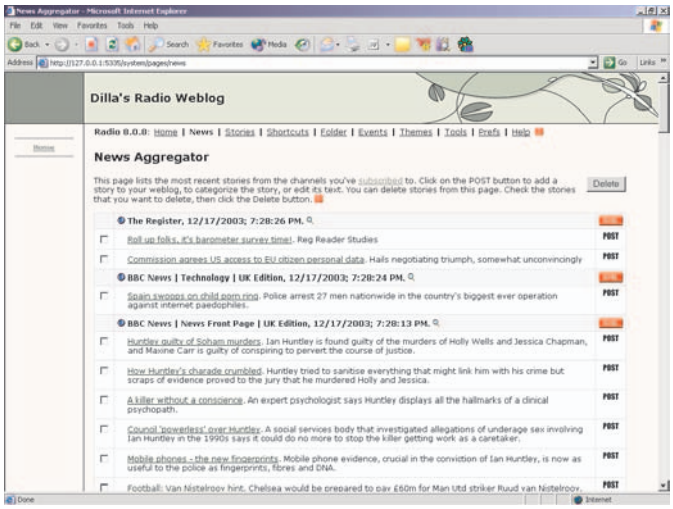
Ini adalah weblog yang disediakan oleh Radio UserLand.



Untuk isi pada *weblog*-nya, Anda dapat menggunakan berbagai macam warna, ukuran, dan jenis *font*-nya.



Anda dapat memilih sumber berita yang Anda inginkan, atau memasukkan alamatnya secara manual.



Ini adalah News Aggregator yang ditawarkan oleh Radio UserLand. News Aggregator ini berbasiskan web.

akan membaca setiap file berita RSS yang beredar di Internet. Sehingga setiap ada file baru, maka secara ototmatis News Aggregator tersebut akan menghimpunnya dalam memperlihatkannya kepada Anda. Berita yang Anda baca melalui News Aggregator ini dapat juga Anda teruskan kepada rekan Anda, atau mengomentarinya dengan menggunakan weblog. Kirimkan saja berita tersebut ke alamat weblog Anda, kemudian tulis komentar Anda. Maka orang lain pun akan ikut dapat membacanya.

Bagaimana Cara Memperolehnya?

Ada beberapa News Aggregator yang tersedia secara gratis ada juga yang berlangganan. Seperti halnya e-mail, News Aggregator juga ada yang berbasis web atau ada juga yang merupakan aplikasi *desktop*. Bahkan ada beberapa News Aggregator yang juga dapat Anda manfaatkan sebagai e-mail Anda.

Bagaimana Cara Menggunakannya?

Untuk mengetahui bagaimana cara memperolehnya secara detail dan bagaimana menggunakannya, maka perhatikan saja artikel "Memiliki Agen Berita Sendiri".

Weblog

Pada bagian tentang News Aggregator sempat disinggung mengenai *weblog*. Dapat dikatakan bahwa weblog adalah yang lebih dulu muncul sebelum News Aggregator sendiri. Kehadiran weblog sendiri telah banyak membantu para jurnalis baik *online* atau bukan untuk menyerahkan beritanya.

Apa yang Dimaksud dengan Weblog?

Kebalikan dari News Aggregator. Jika News Aggregator digunakan untuk membaca, maka weblog digunakan untuk menulis, baik itu berita atau informasi lainnya. Dengan kata lain, weblog adalah catatan seseoang yang sering di-*update* (setiap hari) dan ditujukan untuk dibaca secara umum. Isi weblog sendiri selain dapat berupa berita, juga dapat berisi pendapat dari pemiliknya terhadap sesuatu hal. Dan orang yang menulis weblog dinamakan *bloger*.

Mengapa Menggunakan Weblog?

Karena weblog dianggap lebih efektif ketimbang e-mail. Anda tidak perlu mengetikkan e-mail kepada teman Anda satu per satu. Dengan weblog, secara otomatis jurnal Anda dapat dibaca oleh orang lain. Hal ini akan sangat bermanfaat jika Anda bekerja sebagai wartawan. Jika tidak, weblog tetap saja dapat digunakan untuk menginformasikan pada masyarakat apa yang Anda ketahui atau bagaimana pendapatan Anda terhadap sesuatu hal. Selain itu, Anda juga dapat lebih mudah meng-*update*-nya dibandingkan meng-update sebuah situs.

Dengan weblog, Anda juga tidak perlu menjadi seorang ahli programer, karena dengan *browser* yang bersifat WYSIWYG (*What You See Is What You Get*) akan lebih mempermudah Anda memasukkan berita/informasinya.

Bagaimana Cara Kerjanya?

Data yang Anda isi ke dalam sebuah weblog akan disimpan pada subdirektori. Penyimpanannya menggunakan format RSS. Sehingga bagi orang lain yang mendaftarkan alamat subdirektori tersebut akan dapat mengakses data yang dimasukkan dengan aplkasi atau program yang dinamakan News Aggregator atau Content Aggregator.

Bagaimana Cara Memperolehnya?

Sama halnya dengan News Aggregator, Weblog juga ada yang dapat diperoleh secara cuma-cuma dan ada juga yang mengharuskan Anda berlangganan untuk dapat menggunakannya. Sedangkan basisnya juga hampir sama, weblog ada yang berbasis web dan ada juga yang berbasis desktop. Biasanya yang berbasis dekstop-lah yang harus bayar. Sedangkan untuk yang berbasis web salah satunya yang gratis adalah *radio.weblogs.com*. Untuk mendapatkan weblog dari situs ini, Anda harus melakukan pendaftaran.

Bagaimana Cara Menggunakannya?

Pada halaman weblog Anda, akan disediakan satu buah layar tempat di mana Anda dapat mengisikan berita atau informasi yang ingin dipublikasikan.

Seperti yang telah dikatakan, dengan weblog Anda dapat menuliskan informasi

yang bersifat WYSIWYG sehingga caranya tidak jauh berbeda dengan menulis pada Word Processor. Contoh saja dengan weblog yang dimiliki Radio Userland. Jika Anda Ingin membuat teks tebal, tekan saja tombol *Bold* atau jika ingin mewarnainya dengan berbagai macam warna, pilih warnanya pada *box Color*, dan seterusnya.

Bagaimana Cara Orang Lain Membacanya?

Dengan mendaftar pada situs weblog Anda ditempatkan, maka secara otomatis Anda akan memiliki subdirektori yang dapat dengan bebas Anda berikan atau publikasikan pada semua orang yang Anda kenal. Dan biarkan mereka mendaftarkan alamat weblog Anda tersebuat ke *content aggregator*-nya.

RSS

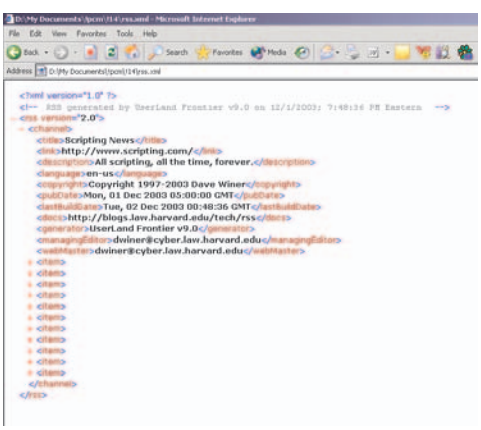
Kedua dasar bahasa yang digunakan, baik dalam News Agrregator maupun Weblog adalah RSS. RSS sendiri merupakan bagian dari XML.

Apa yang Dimaksud dengan RSS?

Jika didefinisikan RSS belum memiliki definisi yang pasti. Sama halnya dengan istilah-istilah lain dalam RSS sendiri. Dan biasanya untuk setiap versi RSS memiliki sebutannya sendiri.

Namun, satu hal yang sudah pasti RSS adalah bagian dari XML yang digunakan untuk mendistribusikan *headline* informasi. Informasi ini sendiri dapat berupa berita, pendapat, atau hanya sekadar pemberitahuan.

RSS yang kali pertama diperkenalkan oleh Netscape ini, kini telah banyak dipergunakan oleh situs-situs terkenal untuk saling berbagi berita, seperti CNN, Disney, CNET, BBC, dan masih banyak lagi.



Anda dapat memiliki berita dari mana saja yang ingin dibaca dengan News Aggregator.

Bentuk File RSS

Ada empat bagian utama dalam sebuah file RSS yang biasa dinamakan RSS feed. Empat bagian itu adalah *channel*, *image*, *item*, dan *text input*.

- ***Channel***: Bagian ini memuat informasi mengenai channel itu sendiri. Seperti siapa pembuatnya dan tentang atau untuk apa channel tersebut. Berikut ini adalah contoh bagian channel:

```
<channel>
<title>MozillaZine</title>
<link>http://
mozillazine.org</link>
<description>Your source for
Mozilla news, advocacy,
interviews,builds,
                        and
more! <description>
<language>en-us</language>
</channel>
```

- ***Item***: Setelah channel, item menjadi bagian penting berikutnya. Item inilah tempat *headline* berita disertakan. Dalam sebuah item, Anda tidak dapat menyertakan judul lebih dari 100 karater dan deskripsi yang lebih dari 500 karakter. Sedangkan dalam setiap channel, file RSS mampu menampung 15 item sekaligus.

```
<item>
<title>Java 2 in Navigator
5?</title>
<link>http://
www.mozillazine.org/
talkback.html?article=607</
link>
<description>Will Java2 be an
integrated part of Navigator
5?
              Read more about
it in this discution…</
description>
```

- ***Image***: Ini adalah bagian untuk memasukkan file gambar. Bagian ini informasinya bersifat opsional. Artinya bagian ini boleh digunakan juga boleh tidak. Jika ingin digunakan ada beberapa aturan yang harus diikuti, mengingat ruang yang tersedia untuk *image* ini tidak besar, hanya cukup untuk seukuran logo saja. Batas minimum pixel-nya dalah 88 pixel (untuk lebar) dan 31 pixel (untuk tinggi), sedangkan ukuran maksimalnya dapat mencapai 400x144 pixel.

```
<image>
<title>MozillaZine</title>
<url>http://
www.mozillazine.org/image/
mynetscape88.gif</url>
<link>http://
www.mozillazine.org</link>
<width>88</width>
<height>31</height>
</image>
```

- ***Text Input***: Sama halnya dengan image, text input juga termasuk bagian yang opsional. Biasanya text input memiliki *hyper link* yang memungkinkan orang yang membaca berita dapat mengirimkan komentarnya. Setiap channel yang diperbolehkan memiliki satu text input saja. Cara penulisannya sebagai berikut:

```
<textinput>
<title>Send</title>
<description>Comments about
MozillaZine?</description>
<name>response Text</name>
<link>http://
www.mozillazine.org/cgi-bin/
sampleonly.cgi</link>
</textinput>
```

Manfaat Lain dari RSS

Selain dapat dipergunakan dalam dunia publikasi, RSS juga dapat dipergunakan untuk *search engine*. Search engine yang berbasiskan RSS feed, memiliki satu keuntungan, yaitu setiap datanya sudah berada dalam keadaan terindeks. Berbeda dengan yang tidak menggunakannya.

Bagaimana Anda Tahu, Situs Tersebut Memuat RSS Feed?

Biasanya untuk situs yang memuat RSS feed terdapat sebuah logo yang bertuliskan XML atau RSS. Sehingga Anda tidak akan bingung apakah situs tersebut memuat file RSS atau tidak. ■

Pengenalan Cahaya untuk Memotret

Fadilla Mutiarawati

Memotret adalah melukis dengan cahaya. Oleh karena itu, cahaya memegang peranan yang sangat penting dalam memotret. Banyak kesalahan yang terjadi karena pencahayan yang salah atau kurang tepat. Sehingga mengetahui teknik pencahayaan menjadi sesuatu yang sangat mendasar, yang harus diketahui oleh seseorang yang ingin memperoleh hasil foto yang baik.

Ada dua sumber cahaya yang kita kenal dalam kehidupan sehari-hari, yaitu cahaya buatan atau biasa yang dikenal dengan lampu dan cahaya alami, seperti matahari. Untuk cahaya buatan, Anda memiliki kemampuan untuk mengatur besar-kecilnya (intensitas) dari cahaya tersebut. Sedangkan untuk cahaya alami, Anda tidak akan mungkin mengaturnya sebagaimana yang Anda inginkan. Namun, bukan berarti yang tidak dapat diatur ini menghasilkan gambar yang kurang baik. Justru sebaliknya, cahaya alami dapat menghasilkan gambar yang sangat baik, jika tahu bagaimana cara memanfaatkannya.

Artikel ini tidak akan memberitahukan Anda posisi bagaimana yang tepat untuk mendapatkan cahaya yang terbaik. Namun, artikel ini akan memberikan pandangan pada Anda bagaimana hasil yang akan diperoleh jika Anda berada pada momen atau posisi tertentu. Sehingga Anda dapat menyesuaikan dengan kebutuhan Anda sendiri, kapan dan posisi bagaimana yang ingin dimiliki untuk foto Anda nantinya.

Letak

Jika objek yang Anda foto menjadi gelap atau kurang terlihat baik, Anda dapat mencoba mengatur posisinya terhadap sumber cahaya yang Anda gunakan. Sebab cahaya yang datang dari atas tidak akan sama dengan cahaya yang datang dari depan.

- **Cahaya dari atas:**

Jika sumber cahaya merupakan matahari, maka posisi ini akan dapat dengan maksimal mengeluarkan warna dari objek yang akan Anda foto. Contohnya seperti pemandangan, bunga, binatang, dan lain sebagainya.

Namun jika cahaya terlalu terang, maka akan membawa bayangan yang sangat gelap dan kontras pada bagian-bagian tertentu yang tertutup sinar. Misalnya, jika memotret wajah seseorang yang menggunakan topi di bawah sinar matahari. Meskipun hal ini juga dapat juga Anda akali dengan menggunakan lampu *flash* dengan modul *fill in flash*. Untuk hal ini akan dibahas lebih lanjut nanti.

- **Cahaya dari depan:**

Cahaya yang datang dari depan biasanya berasal dari lampu *flash* yang ada pada kamera. Pencahaan dengan cara ini adalah yang paling mudah diperoleh, mengingat hampir semua kamera dapat dilengkapi dengan lampu flash. Selain itu, pencahayaan dengan cara ini juga termasuk yang paling ideal. Mengingat dengan cahaya flash, keseluruhan bagian dalam gambar akan mendapatkan cahaya tersebut.

Saking umumnya penggunaan cahaya seperti ini mengakibatkan gambar yang dihasilkan terlihat terlalu umum sehingga terkesan membosankan. Di samping itu, pencahayaan yang berlebihan dari depan juga dapat mengakibatkan kurangnya detail pada gambar.

Pemotretan cahaya dari depan (*flash*), yang sangat umum digunakan.

Jika cahaya datang dari belakang, pemotretan tanpa *flash* dapat membuat gambar menjadi seperti siluet.

Pemotretan yang menggunakan cahaya samping, tidak akan terlihat membosankan dibandingkan dari depan.

Gambar terlihat datar, karena kurangnya bayangan yang justru kadang dapat membuat gambar lebih detail. Dan jika ada *background* yang tidak terkena cahaya, maka *background* tersebut akan terlihat sangat gelap dan kontras.

- **Cahaya dari samping:**

Cahaya yang datang dari samping sangat cocok untuk digunakan dengan modul *portrait*. Sebab dengan memanfaatkan cahaya dari samping, baik tekstur, bentuk, maupun pola dari sebuah objek dapat lebih tajam dan hidup. Jika gambar yang Anda hasilkan memiliki perbedaan yang terlalu kontras, maka cobalah untuk menggunakan sedikit bantuan cahaya *flash*.

- **Cahaya dari belakang:**

Cahaya yang datang dari belakang sangat tepat digunakan untuk membuat foto siluet, atau digunakan untuk mengeluarkan efek cahaya halo dari sebuah *portrait*. Namun cahaya yang datangnya dari belakang jika bukan sebuah kesengajaan, maka akan membuat gambar objek menjadi gelap. Jika diakali dengan mengompensasi diafragma, maka *background* objek akan terlihat terlalu terang.

Cahaya Natural

Seperti yang dikatakan sebelumnya bahwa matahari merupakan salah satu sumber cahaya yang tidak dapat diatur, baik intensitas maupun letaknya. Sehingga bagi Anda yang ingin memanfaatkan matahari sebagai sumber cahaya foto Anda, maka Andalah yang harus menyesuaikan objek foto tersebut dengan matahari.

Tidak ada waktu yang paling baik untuk memotret, sebab masing-masing bagian waktu memiliki keistimewaannya sendiri. Misalnya, memotret pegunungan di pagi hari dengan memotret pantai di pagi hari tentu akan memberikan kesan yang berbeda. Begitu pula dengan memotret pantai di senja hari dengan pegunungan di senja hari, tentu saja akan berbeda juga.

Misalnya di pagi hari, Anda dapat mencoba memotret jatuhnya cahaya pagi (*morning mist*) yang berebutan menembus tumpukan daun pada dahan pohon di atas, atau wajah embun yang memang hanya muncul di pagi hari. Bunga-bunga pun masih terlihat sangat segar di pagi hari.

Pemotretan di siang hari dengan cahaya di atas, akan membuat bayangan gelap pada beberapa sisi (contohnya, pada leher dan sekitar mata).

Gunakanlah *flash* (*fill in flash/auto flash*) untuk menghilangkan sisi gelap yang tidak diinginkan.

Jika objek menggunakan topi atau payung, bayangan akan membuat gambar tidak bagus.

Gunakan juga *flash* (*fill in flash/auto flash*) di siang hari untuk membuat profil objek lebih jelas.

Jika cuaca terasa kurang bersahabat karena sedikit mendung, maka ini adalah saat terbaik Anda untuk membuat potret orang-orang yang Anda kasihi. Sebab cahaya mendung tidak akan menimbulkan bayangan yang terlalu gelap. Seperti pada bawah mata, hidung, dan dagu. Juga untuk bunga-bunga berwarna pastel, cuaca mendung adalah saat yang baik untuk memotretnya dengan tekstur yang lembut.

Bagaimana jika hujan? Jangan takut untuk memotret di kala hujan. Kadang refleksi cahaya lampu yang berwarna-warni pada saat hujan (gerimis) sangat menarik untuk menghias foto Anda. Satu hal yang perlu Anda perhatikan ketika memotret dalam hujan adalah keamanan kamera Anda sendiri dari air hujan. Jangan sampai air hujan tersebut merusak kamera Anda, terutama bagi kamera digital yang sangat rentan terhadap air. Selain itu, perhatikan kompensasi *exposure* yang digunakan. Cobalah untuk menggunakan +0.5 atau +0.1.

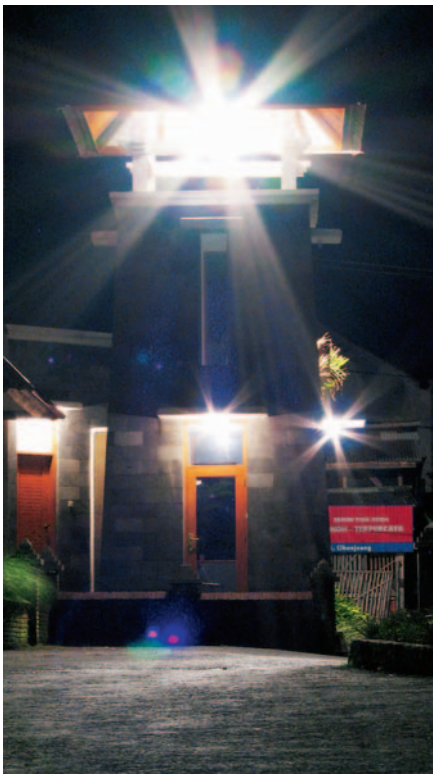


Foto di malam hari tanpa menggunakan *flash*, dengan kecepatan rendah dan bukaan (difragma) yang besar.

Begitu pula jika Anda mencoba memotret dalam kabut. Cobalah lakukan kompensasi *exposure* yang sama dengan memotret pada hujan. Sebab biasanya dalam keadaan ini cahaya akan sangat minim, karena akan terhalang oleh hujan atau kabut di sekililingnya.

'Menunggu' dan 'sabar' adalah kata kuncinya jika ingin mendapatkan gambar yang tepat. Sebab jika kita terburu-buru, maka ekspresi yang diinginkan oleh foto tersebut, bisa tidak Anda dapatkan.

Satu lagi momen yang sangat sayang Anda abaikan adalah pemandangan malam. Tidak jarang karena rendahnya cahaya pada malam hari, banyak foto pemandangan malam menjadi gagal. Padahal jika Anda jeli ada beberapa fitur dalam kamera Anda yang dapat dimanfaatkan untuk mempercantik foto malam tersebut. Untuk lebih jelasnya, Anda dapat membaca boks "Foto di Malam Hari".

Cahaya Lain

Ada beberapa jenis cahaya lain yang dapat Anda manfaatkan agar gambar yang dihasilkan memiliki ekspresi yang diinginkan. Cahaya lain yang akan dibahas di sini adalah cahaya lampu. Pertama cahaya lampu yang umum sekali digunakan, yang cahayanya berwarna agak kekuning-kuningan. Lalu yang kedua adalah cahaya lampu yang menyala seperti lampu neon. Berwarna sangat putih.

Masing-masing cahaya memberikan pengaruh berbeda terhadap gambar yang dihasilkan. Jika warna bohlam yang biasa dikenal *tungsten* ini dapat menimbulkan kesan hangat, maka lampu neon atau yang disebut juga *fluorescent* dapat memberikan kesan yang lain.

Namun jika kamera Anda dilengkapi dengan fitur *white balance* biasanya, baik modul tungsten maupun fluorescent telah disediakan. Sehingga Anda tidak perlu repot mencari lampu bohlam yang banyak untuk menghangatkan gambar Anda. Cukup *setting* kamera Anda dengan white balance yang berada pada modul tungsten. Tungsten biasanya disimbolkan dengan icon lampu bohlam dan fluorescent dengan gambar persegi panjang yang bercahaya.

Dalam hal memotret, kita tidak selalu dapat mengandalkan *mood* dari hanya sedikit cahaya yang digunakan. Sebab kamera tidak dapat menangkap cahaya seperti layaknya mata manusia. Satu bohlam yang terlihat hangat di mata manusia, belum tentu tertangkap oleh kamera. Oleh sebab itu, lakukanlah pengaturan white balance ini meskipun lampu yang Anda gunakan sudah sesuai dengan mood yang diinginkan.

Jika gambar yang Anda ambil sudah terlanjur dalam keadaan normal, maka untuk memberikan efek tungsten dan fluorescent ini, Anda harus melakukannya dengan *software image editing*. Misalnya, dengan Photoshop.

Flash

Jika Anda membeli kamera *pocket*, maka lampu *flash* akan disediakan secara otomatis pada kamera dengan *setting*-an standar. Sedangkan jika membeli kamera profesional, maka Anda dapat menambahkan lampu flash pada bagian atas kamera tersebut. Ada beberapa hal yang perlu diperhatikan dalam menggunakan lampu flash jika menginginkan gambar Anda maksimal, yaitu:

- Perhatikan jarak objek dengan jangkauan yang mampu diraih oleh lampu flash Anda.
- Baterai dari lampu atau kamera juga tidak boleh luput diperhatikan, karena baterai yang tidak maksimal akan membuat cahaya yang dihasilkan juga kurang maksimal.
- Kecepatan juga dapat mempengaruhi jarak foto dengan flash. Semakin cepat, maka jarak dapat semakin jauh. Untuk dapat mengetahui bagaimana menyesuaikan menggunakan lampu flash dengan jarak memotret, cobalah lihat keterangan yang diberikan oleh kamera atau bagian belakang lampu flash Anda.
- Daripada menggunakan fitur *red aye reduction*, ada baiknya jika Anda menerangkan lampu di sekitar objek agar pupil mata mereka mengecil. Sebab biasanya fitur ini agak sedikit mengganggu bagi objek.

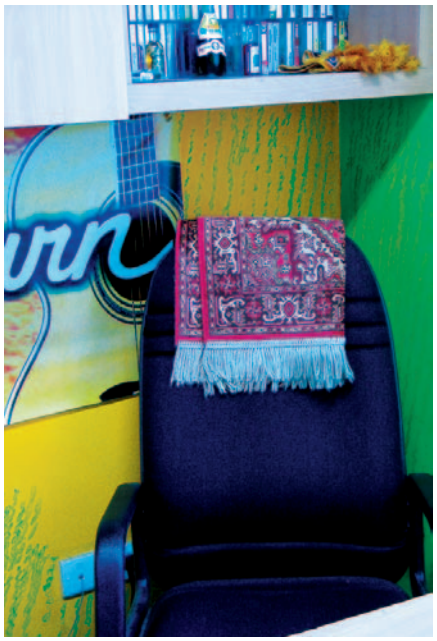
Fill in Flash

Ada salah satu metode dalam menggunakan lampu *flash* yang dikenal dengan sebutan *Fill in Flash*. Metode ini hanya akan memberikan kilatan lampu flash secukupnya pada objek yang Anda foto.

Fill in Flash sangat cocok digunakan untuk memotret seseorang yag menggunakan topi atau payung di siang hari.

Jangan menggunakan *flash* untuk permukaan yang licin seperti layar monitor ini.

Gambar dengan white balance Auto secara otomatis kamera akan lebih condong memilih florescent).



Gambar dengan white balance tungsten..

Gambar dengan white balance florescent.

Modul ini juga sangat cocok untuk memotret seseorang di tengah hari, sekitar jam 12 siang. Fill in Flash ini juga dapat mengantisipasi siluet yang tidak diinginkan berkaitan cahaya yang datang dari belakang.

Tanpa Flash

Namun ada saat-saat tertentu di mana dalam cahaya yang tidak terlalu terang, penggunaan lampu *flash* justru dapat mengurangi keindahaan dari sebuah foto itu sendiri. Misalnya jika memang Anda ingin membuat sebuah gambar siluet dari pemandangan senja atau karena objek yang ingin Anda foto terlalu jauh, maka penggunaan lampu flash akan sia-sia, bahkan dapat mengubah hasil gambar yang diinginkan.

Selain itu, lampu flash juga akan sangat buruk digunakan jika di sekitar gambar yang akan Anda ambil terdapat sebuah benda mengkilap yang dapat memantulkan cahaya dari lampu flash itu sendiri, seperti kaca dan genangan air. Atau bisa juga pada saat memang lampu flash tidak boleh digunakan, seperti dalam sebuah acara atau gedung. Jika hal ini terjadi maka, Anda harus mencari posisi yang tepat (objek mendapatkan cahaya seperlunya) lalu ambillah gambar dengan keadaan yang sangat stabil dengan meningkatkan *exposure*.

Memotret di Malam Hari

Memotret di malam hari memang tidak dapat dikatakan mudah. Selain itu, setiap kamera punya cirinya masing-masing, sehingga Anda harus banyak melakukan percobaan dengan jenis kamera yang sama secara berkali-kali untuk mendapatkan hasil yang diinginkan.

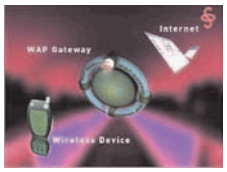
Untuk kamera *compact*, biasanya penyetelannya untuk memotret malam sangat terbatas sehingga gambar yang dihasilkan kurang maksimal. Jika gambar yang diambil berupa *portrait*, maka hanya akan ada dua pilihan. Jika menggunakan flash, objek akan terlihat jelas, namun *background* akan terlihat gelap. Sedangkan jika tidak menggunakan flash gambar akan terlihat gelap.

Lain halnya dalam mengambil gambar pemandangan, maka gambar yang dihasilkan akan sangat gelap. Sebab gambar pemandangan tidak mungkin menggunakan flash. Sedangkan untuk exposure kamera *pocket* tidak memiliki pilihan. Padahal banyak hal yang dapat dieksplorisasi dalam memotret di malam hari. Namun, keterbatasan ini dapat sedikit diakali dengan menggunakan pengaturan ISO yang tinggi. Jangan terlalu tinggi juga, sebab walaupun ISO tinggi dapat membuat gambar lebih terang. ISO yang tinggi juga memuat *noise* yang lebih banyak/besar. Lain hasilnya jika Anda menggunakan kamera yang dilengkapi fitur lebih banyak. Foto yang dihasilkan pun dapat lebih bervariasi.

Yang perlu Anda perhatikan jika ingin memotret pemandangan malam adalah gunakan selalu ISO rendah antara 50-100. Kemudian gunakan pengaturan foto yang bergantung kepada kecepatan, biasanya modul ini dilambangkan dengan huruf S (*Shutter Priority*) atau *Tv Time Value*. Cobalah untuk mengambil foto pada kecepatan antara 1-15 detik. Kemudian bandingkan hasilnya. Anda dapat memilih mana yang Anda sukai. Sedangkan untuk diafragmanya cobalah untuk menggunakan difragma antara f10-f8. Jangan gunakan lampu flash Anda, sebab selain tidak efektif akan merusak gambar yang Anda inginkan. Lampu malam dapat lebih terlihat natural juga jika Anda menggunakan pengaturan *white balance* pada modul tungsten.

Satu hal yang tidak boleh dilupakan jika ingin memotret pemandangan malam, yaitu tripod! Rendahnya kecepatan yang Anda gunakan sangat sulit Anda lakukan jika tidak menggunakan tripod. Oleh sebab itu, tripod sangat penting pada pemotretan malam. Dan yang terakhir adalah jangan terlalu bergantung kepada gambar yang LCD perlihatkan. Sebab pengaruh backlit pada LCD akan membuat gambar terlihat lebih terang, tidak sebagaimana aslinya. ■

Mengenal Wireless Application Protocol (WAP) dengan Lebih Baik



Wisnu A. Prastyanto

Wireless Application Protocol **(WAP) bukanlah sesuatu yang baru dewasa ini. Banyak sekali alat komunikasi yang telah menyertakan teknologi tersebut ke dalam alat mereka, tentunya dengan tujuan untuk dapat memberikan pelayanan lebih kepada penggunanya. Dan banyak sekali yang dapat kita peroleh dengan menggunakan teknologi tersebut.**

Mungkin Anda salah satu pengguna ponsel yang sudah memiliki teknologi WAP di dalamnya, tetapi sayangnya Anda hanya bisa menggunakan teknologi tersebut tanpa mengetahui dengan benar apakah teknologi WAP tersebut sebenarnya. Karena saat ini banyak sekali alat komunikasi *mobile* yang telah menyediakan fasilitas tersebut, tentunya setelah bekerja sama dengan penyedia layanan TI-nya.

WAP adalah sebuah lingkungan aplikasi dan kumpulan protokol komunikasi yang diperuntukkan bagi alat-alat tanpa kabel yang memungkinkan pabrikan, vendor TI, dan pengguna alat tersebut untuk dapat mengakses ke dalam Internet dan pelayanan *telephony* lainnya. Dengan WAP, jurang antara dunia *mobile* dan Internet dapat dihilangkan, karena WAP mampu membawa informasi secara *online* melewati internet, seperti *mobile banking*, e-mail, secara langsung ke dalam ponsel yang Anda miliki.

Komponen Arsitektur WAP

Berikut ini beberapa keterangan singkat mengenai komponen arsitektur WAP yang telah digambarkan di atas:

Wireless Application Environment (WAE)

Wireless Application Environment berfungsi untuk menggabungkan World Wide Web (WWW) dengan teknologi yang dimiliki oleh ponsel. Objek pokok yang diterapkan WAE ini adalah mengatur operasi-operasi yang telah diatur oleh operator (vendor) dan penyedia layanan untuk membangun aplikasi dan layanan yang dapat digunakan melalui *platforms wireless* yang berbeda sehingga menghasilkan efisiensi dan data yang cukup bermanfaat bagi penggunanya. Di dalam WAE berisi *micro browser* yang berfungsi sebagai:

- Pendukung *Wireless Markup Language* (WML, bahasa pemrograman yang digunakan pada teknologi mobile).
- Pendukung WMLscript, penggunaannya hampir sama dengan JavaScript.
- Pendukung *interface* pemrograman dan layanan telephony yang berisi format data, gambar, dan lain-lain.

Wireless Session Protocol (WSP)

Wireless Session Protocol (WSP) berfungsi sebagai pembuka atau mengakhiri suatu koneksi jaringan. WSP sendiri terbagi menjadi dua buah *session*, yaitu *connection-oriented*, koneksi yang khusus untuk berinteraksi dengan operasi yang terjadi pada *Transaction Layer Protocol* (WTP) dan yang kedua adalah *Wireless Datagram Protocol* (WDP), koneksi yang berkaitan erat dengan keamanan data. Selain hal tersebut WSP juga mendukung:

- Fungsi dari HTTP/1.1 serta juga pengodean data.
- Pemeriksaan data yang *reliable* dan *unreliable*.
- Pengontrolan waktu akses data.

Wireless Transaction Protocol (WTP)

Wireless Transaction Protocol (WTP) berfungsi memeriksa format data, konversi data dan atau pengodean yang akan ditransmisikan, misalnya:

- Melakukan pemeriksaan keabsahan data antarpengguna.
- Pemeriksaan terhadap nomor pesan yang akan dikirim.
- Melakukan sinkronisasi data transaksi.

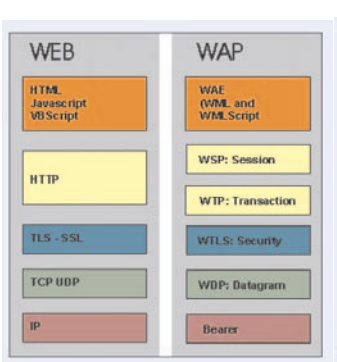
Wireless Transport Layer Security (WTLS)

Wireless Transport Layer Security (WTLS) adalah sebuah protokol yang berfungsi sebagai keamanan data yang disesuaikan oleh standar industri *Transport Layer Security* (TLS) yang tentunya mendukung *Secure Sockets Layer* (SSL). WTLS tersebut ditujukan pada penggunaan aplikasi WAP, yang terdiri dari:

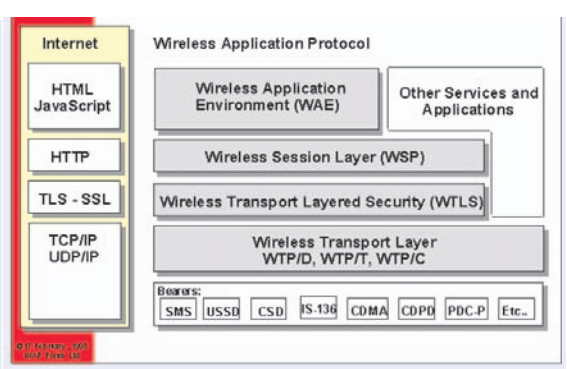
- Integritas data, mengenai keabsahan dari data yang akan dikirim.
- *Privacy*, menyangkut kerahasiaan data, di mana data yang akan dikirim tidak akan bisa didapat oleh orang lain yang tidak bersangkutan.
- *Authentication*, menyangkut kebenaran jati diri pengguna.
- *Denial-Of-service protection*, pemeriksaan dan pembuktian pengamanan data yang valid.

Wireless Datagram Protocol (WDP)

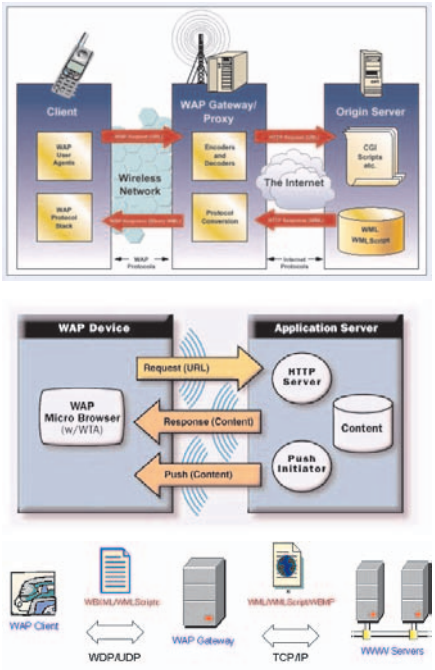
Wireless Datagram Protocol (WDP)



Perbedaan arsitektur WWW dan WAP.



Komponen arsitektur *Wireless Application Protocol* (WAP).



Skema dari model WAP.

merupakan kelanjutan dari WTLS yang mampu berkomunikasi dengan *bearer*. Bertugas untuk melakukan transmisi data dalam format *biner* melalui media *gateway*, serta melakukan definisi pengalamatan jaringan yang nantinya akan dapat dikenalai oleh lapisan bearer.

Bearers

Bearers itu sendiri terdiri atas data *switch, short message*, dan data paket. Ia memiliki fungsi untuk dapat melakukan transfer data dari suatu unit informasi yang berisi alamat dan melakukan pemeriksaan kesalahan (*error checking*) serta melakukan penundaan transfer hingga proses yang dilakukan sudah benar.

Others Services and Applications

Selain dari *layer-layer* yang telah dijelaskan, ada beberapa pendukung aplikasi lainnya yang merupakan program aplikasi jaringan, seperti e-mail, *notepad, phonebook*, dan lain-lain. Kesemuanya itu diberikan agar pengguna dapat merasakan manfaat yang cukup banyak.

Wireless Application Protocol (WAP) Model

Model dari teknologi WAP sebenarnya tidak terlalu berbeda jauh dengan model World Wide Web (WWW) yang telah ada sebelumnya, karena pada dasarnya WAP juga menggunakan standar komunikasi protokol pada WWW tersebut.

Prinsip kerjanya adalah aliran data dari pengguna WAP akan mengirimkan *encoded request*. Kemudian protokol gateway akan mentranslasikan (menterjemahkan) request dari WAP protokol yang terdiri dari WSP, WTP, WTLS, dan WDP tersebut menuju WWW protokol (*origin server*, yaitu HTTP, TCP/IP). *Encoder* kemudian akan melakukan penyesuaian format data dengan server jaringan WWW yang dapat berupa CGI dan Script, di mana nantinya server akan merespon *request* tersebut dan kemudian akan mengirimkannya kembali melalui protokol gateway untuk kemudian diterjemahkan kembali menuju WAP pengguna yang dalam hal ini adalah pengguna ponsel tersebut.

Manfaat WAP

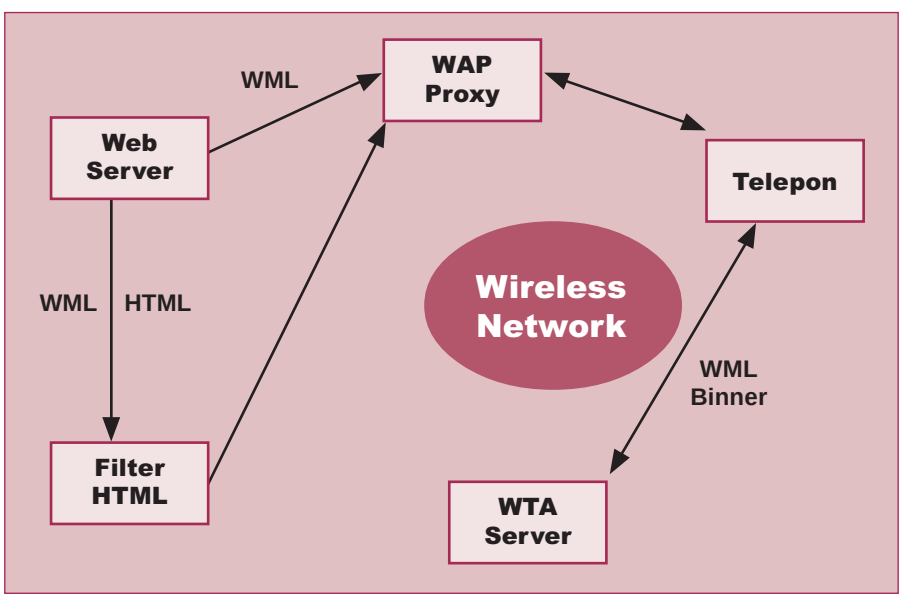
Jika kita membicarakan manfaat, maka harus dilihat dari beberapa sisi. Dari sisi pengguna, vendor TI, atau pengembang teknologi WAP itu sendiri. Apabila kita melihat dari sisi pengguna WAP, maka keuntungan yang dapat dirasakan adalah mudah dan tingkat keamanan yang lebih dalam mengakses Internet dan kemampuan untuk dapat mengakses pelayanan lainnya.

Tetapi jika dari sisi pengembang teknologi WAP itu sendiri, maka keuntungan yang dapat dirasakan adalah bahwa banyak alat yang akan dapat mengakses apa yang mereka ciptakan dengan begitu akan lebih banyak pasar yang dapat diraih. Jadi pada prinsipnya, banyak keuntungan yang dapat dioperoleh oleh banyak pihak dengan menggunakan teknologi WAP tersebut.

Contoh Kasus Jaringan WAP

Dalam contoh ini, komunikasi WAP pengguna memiliki dua buah server jaringan *wireless*, yaitu terdiri atas WAP proxy yang berfungsi untuk mentranslasikan WAP request menuju WWW *request web server*. Proxy tersebut juga melakukan pengodean dari web server untuk kemudian menerjemahkan format *biner* sehingga dapat dimengerti oleh pengguna.

Jika web server mengirimkan data dalam format WML, WAP proxy akan langsung mengirimkan ke media ponsel, dan jika web server mengirimkan data berupa HTML, maka akan disaring terlebih dahulu oleh HTML filter kemudian akan diterjemahkan menjadi file WML sehingga akan dapat dimengerti oleh WAP proxy server dan langsung dikirimkan menuju ponsel. Sedangkan WTA server (*Wireless Telephony Application*) adalah sebuah server gateway yang akan merespon request dari WAP *client* secara langsung. ■



Contoh kasus jaringan WAP.

Hati-hati dengan Maling!

Fadilla Mutiarawati

Memperhatikan kemanan toko Anda sangat penting! Sebab jika tidak, maka pengutil akan bersarang di toko Anda.

Pernahkah Anda berjalan keluar dari kasir, lalu ada alarm yang berbunyi? Jika ya, itu tandanya dalam belanjaan Anda ada yang belum terhitung atau ada alat pengaman dari barang belanjaan Anda yang belum dilepaskan.

Seperti yang tadi disebutkan, ada berbagai macam hal yang dapat membuat alarm supermarket atau *department store* berbunyi. Itu tandanya ada lebih dari satu sistem yang digunakan oleh pusat perbelanjaan saat ini.

Jika pada edisi sebelumnya telah dibahas mengenai sebuah teknologi baru yang dinamakan RFID, maka pada edisi kali ini pembahasannya akan lebih meluas bukan pada teknologi label atau tag yang digunakan tapi sistem kemanan yang diterapkan. Semoga saja artikel ini dapat membantu mereka yang ingin membuka sebuah toko atau sekadar ingin mengetahuinya.

Sistem Keamanan

Besarnya angka pencurian yang terjadi dari hari ke hari, mengakibatkan banyak pasar swalayan harus mengeluarkan dana lebih untuk mengamankan tokonya. Bayangkan saja pada beberapa tahun lalu jumlah pengutilan di pusat perbelanjaan Amerika Serikat sempat mencapai angka US$26 miliar (*www.howstuffworks.com*). Dan hal ini akan meningkat terus jika sistem keamanan tidak kunjung diperbaiki.

Besarnya kebutuhan terhadap sistem keamanan ini juga telah menumbuhkan sebuah industri baru yang memang khusus berkecimpung dalam bidang ini. Jika dibagi menurut metodenya, ada tiga macam metode yang dapat diterapkan. Mulai dengan cara mengamati sebagai metode paling kuno, lalu menampilkannya dalam *display* sampai dengan cara yang paling canggih yaitu dengan memberikan tag khusus dan sistem alarm.

Metode 1, Mengamati

Mengamati pelanggan Anda adalah metode yang paling lama yang pernah diterapkan sebagai sistem kemanan dalam pasar swalayan. Namun jika kali pertama metode ini 100% menggunakan tenaga manusia, sekarang sebagian manusianya telah tergantikan dengan kamera pengintai. Biasanya kamera-kamera ini diletakkan pada lorong-lorong produk dan monitor pengamatnya berada pada ruangan tersendiri.

Sampai saat ini, teknologi pengamatan dengan kamera masih digunakan. Tetapi bukan lagi hanya dalam pasar swalayan melainkan dalam gedung perkantoran, rumah sakit, apartemen, sampai pada rumah-rumah besar atau rumah orang-orang penting.

Bagi supermarket atau department store memasang kamera pengintai, bukan lagi untuk mengamati calon konsumen yang mencurigakan. Melainkan untuk file dokumentasi, jika suatu saat terjadi sesuatu. Dengan memiliki rekaman gambar, setiap tempat perbelanjaan dapat selalu mengevaluasi keamanan lokasi bisnisnya.

Oleh karena itu, teknologi kameranya pun terus berkembang. Pada awalnya bentuk kamera pengintai lumayan besar. Namun sekarang kamera pengintai yang sangat kecil sudah banyak tersedia di mana-mana dan dijual bebas. Begitu pula media transmisinya, yang kini sudah tersedia dengan menggunakan gelombang radio.

Hasil pengamatannya juga dapat Anda lakukan di mana saja, baik melalui layar monitor yang langsung dihubungkan atau melalui komputer di mana saja Anda berada. Jika ingin mengamati hasil intaian secara *online*, maka Anda harus menghubungkannya juga ke Internet.

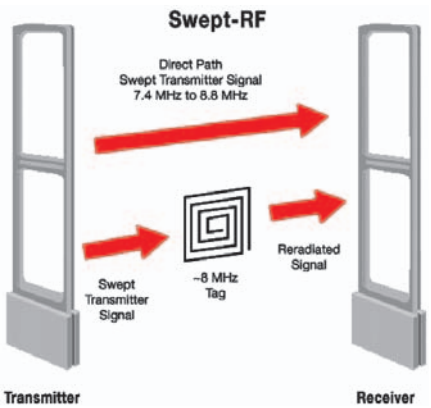
Metode 2, Diamankan

Pada metode kedua ini ada dua pilihan. Yang pertama membuat *counter* untuk setiap ketegori barang yang dijual. Dan yang kedua adalah dengan mengamankan setiap barang dengan seutas tali baja yang dikaitkan ke rak barang.

Untuk cara yang pertama akan sangat merepotkan. Karena selain memakan waktu penjualan, cara ini juga memakan banyak tenaga. Kecuali untuk toko yang sangat kecil dan tidak terlalu ramai. Sedangkan untuk diterapkan pada supermarket atau *department store* yang besar dan ramai, cara ini tidak akan efektif dan efisien. Apalagi untuk produk-produk retail yang kadang harus dicoba dulu oleh pembelinya. Atau produk-produk makanan segar yang setiap waktu harus diganti dengan yang lebih segar lagi.

Metode pengaman yang seperti ini cocoknya hanya diterapkan pada toko yang menjual barang-barang elektronik. Itupun akan membuat konsumen sedikit kurang nyaman, sebab tidak dapat mengamati lebih jauh produk yang akan dibelinya. Dengan cara seperti seorang petugas, jaganya pun harus selalu siap membuka kunci dan *standby* melayani kustomer yang ingin melihat barang tersebut lebih teliti lagi.

Sampai saat ini cara tersebut juga masih digunakan, meskipun ikut ditemani dengan metode yang pertama. Sedangkan tali yang digunakan pada



Kerja sistem alarm yang yang menggunakan radio frekuensi .

metode ini bukanlah tali biasa. Tali tersebut kebanyakan terbuat dari logam, sehingga akan sangat sulit untuk diputuskan atau ditanggalkan. Bahkan semakin hari kekuatan tali ini dikembangkan. Mulai dari bahan tembaga biasa, sampai pada tali yang terbuat dari baja juga sudah mulai dipergunakan.

Satu lagi teknologi yang kadang diterapkan pada metode ini adalah penggunaan alarm yang dapat berbunyi jika tali putus atau lepas secara paksa.

Metode 3, Label/Tag dan Alarm

Ini adalah metode yang paling populer dan banyak digunakan di masa sekarang. Sistem alarm memang lebih mahal, namun jauh lebih efektif. Di samping itu, sistem ini tidak akan mengganggu kenyamanan konsumen yang sedang berbelanja. Setiap konsumen tidak akan merasa seperti sedang diawasi dan dapat bebas mengeksplorasi barang yang akan dibelinya. Sistem ini dinamakan juga *Electronic Article Surveillance* (EAS).

Bagaimana Cara Kerja Sistem Alarm?

Jika ada sebuah barang yang telah memiliki sebuah label atau tag keluar tanpa sepengetahuan mesin kasir atau tanpa ditanggalkan terlebih dahulu, maka alarm akan berbunyi. Secara spesifik, metode ini memiliki cara kerja yang beragam tergantung pada sistem alarm yang digunakan.

Pada saat ini sudah ada tiga sistem yang dipergunakan, yaitu sistem yang menggunakan radio frekuensi, lalu sistem yang menggunakan elektromagnetik, dan terakhir dan yang paling baru dan banyak dipergunakan adalah sistem dengan *acousto-magnetic*. Berikut penjelasan masing-masing sistem.

Seperti yang sempat disinggung sebelumnya, bahwa setiap barang harus memiliki label atau tag agar sistem ini dapat berfungsi. Yang membedakan dari satu sistem alarm ke alarm lainnya adalah bagaimana sebuah tag atau label dibaca oleh *reader*-nya.

Masing-masing selain memiliki cara kerja yang sedikit, ketiganya juga menggunakan frekuensi yang berbeda-beda, sehingga jarang gerbang dari satu sistem ke sistem yang lain pun lebih bervariasi. Tergantung pada frekuensi yang dipergunakan. Jika frekuensi semakin besar, maka jarak antargerbang pun dapat semakin jauh. Yang dimaksud dengan gerbang ini adalah dua alat yang menyerupai pembatas yang berfungsi masing-masing sebagai *transmitter* dan *receiver*.

Sistem yang berbeda juga akan mempengaruhi bahan yang digunakan untuk tag atau label. Sehingga masing-masing sistem memiliki kelebihan dan kekurangannya masing-masing.

Radio Frekuensi

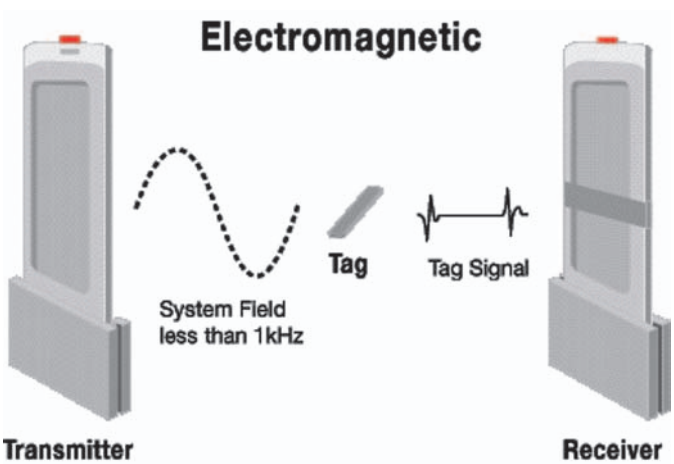
Sistem radio frekuensi menggunakan signal radio berfrekuensi 2 MHz–10 MHz. Dan jarak gerbang (antara transmitter dengan receiver) dapat mencapai 80In. Sistem ini adalah yang paling banyak digunakan di Amerika. Ketika sebuah tag berjalan melewati gerbang, maka transmitter akan mengirimkan sinyal radio pada tag. Kemudian tag akan memprosesnya dan mengirimkan kembali ke receiver. Jika receiver dapat menerima sinyal balik tersebut, maka alarm akan berbunyi. Oleh karena itu, untuk membuat barang dapat melewati gerbang tanpa membunyikan alarm tag atau label harus terlebih dahulu dinonaktifkan dengan cara memberikannya tegangan melalui frekuensi yang jauh lebih tinggi sehingga diode yang ada dalam tag atau label terbakar. Jika diode terbakar, maka tag/label dapat melewati gerbang tanpa harus membunyikan alarm.

Elektromagnetik

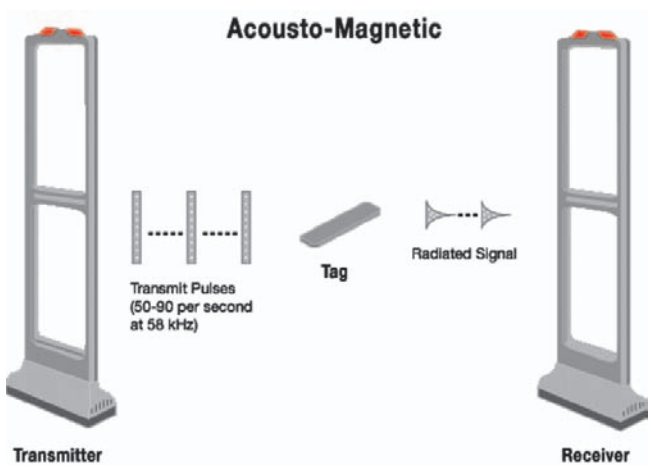
Sistem yang paling banyak digunakan di Eropa ini menggunakan sinyal elektromagnetik yang berfrekuensi antara 70 Hz sampai 1 KHz. Dan jarang antara transmitter dengan receiver-nya (gerbang) dapat mencapai 40In.

Ketika sebuah tag/label melewati gerbang dengan sistem elektromagnetik, maka transmitter akan mengirimkan sinyal elektromagnetik pada tag/label. Kemudian label akan memprosesnya dan mengirimkan kembali sinyal tersebut ke receiver. Sinyal yang dikirimkan kembali ke receiver pada sistem ini ada dua pola. Pola yang pertama akan membuat alarm berbunyi, sedangkan pola yang kedua adalah pola yang menandakan barang layak untuk keluar (alarm tidak akan berbunyi).

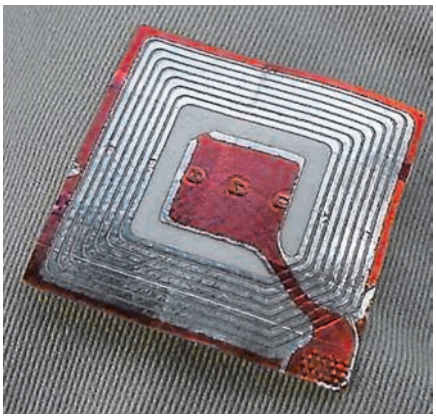
Tag/label pada sistem ini dapat diaktifkan kembali setelah digunakan. Atau dapat juga dikatakan bahwa sistem ini dapat menggunakan *reusable tag/label*. Oleh karena itu, sistem ini



Kerja sistem alarm yang menggunakan elektromagnet.



Kerja sistem alarm yang menggunakan a*cousto-magnetic*.


Contoh bagian dalam sebuah tag RFID.

Berbagai macam jenis tag.

Kamera pengintai ini dikenal juga dengan nama *Close Circuit Television* (CCTV).

banyak juga digunakan pada perpustakaan.

Acousto-magnetic
Frekuensi yang digunakan oleh sistem ini adalah ±58 KHz. Ketika tag telah mendapatkan keseluruhan sinyal dari transmitter, maka sinyal yang ada pada tag/label akan dideteksi oleh receivernya. Frekuensi yang terdeteksi oleh receiver akan dikonfirmasikan kembali ke transmitter. Jika keadaan ini sama, maka alarm akan berbunyi. Namun jika tidak, itu tandanya barang layak keluar.

Selain ketiga sistem ini, sebenarnya masih ada satu sistem yang juga digunakan dalam hal menjaga keamanan perbelanjaan, yaitu sistem yang menggunakan gelombang microwave. Ini adalah salah satu sistem yang tertua pada metode ketiga (EAS). Sistem ini masih memiliki nilai akurasi 80%, sehingga masih ada beberapa toko yang sampai saat ini masih juga menggunakannya.

Jenis tag atau label sendiri sebenarnya ada beberapa macam. Selain RFID yang Anda kenal lewat edisi yang lalu, masih ada lagi beberapa jenis label atau tag yang dipergunakan pada supermarket atau *department store*.

- **Disposable tag**

Ini adalah tag yang kapan saja dapat dihancurkan. Biasanya tag atau label ini terbuat dari kertas. Tag atau label ini tidak hanya dapat dilengkapi dengan *barcode*, tetapi juga informasi lainnya seperti nama dan harga barang.

Disposable tag tidak hanya dapat ditempatkan pada benda yang besar, untuk benda-benda kecil seperti anting dan sebagainya tersedia juga disposable tag khusus yang berukuran sangat kecil dan tidak akan mengganggu kenyamanan konsumen ketika sedang mengamati benda tersebut.

- **Reusable tag**

Reusable tag adalah tag yang dapat digunakan berkali-kali. Contohnya saja tag yang digunakan oleh Carrefour Supermarket untuk produk retailnya seperti pakaian dan sepatu.

Reusable tag terbuat dari bahan plastik dan tidak mudah untuk dilepaskan. Untuk melepaskan tag tersebut, Anda harus menggunakan sebuah alat khusus yang dinamakan *detacher*. Sehingga, meskipun Anda berhasil membawa pulang produk—dengan tag yang masih menempel—tanpa membunyikan alarm, tetap saja Anda tidak akan bisa membuka tag ini, tanpa merusak produk tersebut.

- **Benefit Denial tag (tag yang merugikan)**

Ada salah satu bentuk tag yang terbuat dari plastik yang berisikan cairan. Cairan ini berupa tinta yang tidak akan mungkin dapat luntur kembali. Sehingga jika Anda memaksakan untuk membukanya sendiri, maka Anda maupun produk tersebut akan berada dalam masalah. Di samping produk tersebut tidak akan mungkin dapat dimanfaatkan (baik dipakai maupun dijual kembali), Anda pun harus bersusah payah menghapus tinta yang mengotori tangan Anda.

Selain jenis tag dalam sistem alarm juga ada beberapa macam alat juga yang dipergunakan untuk mencopot atau menonaktifkan label/tag tersebut.

- **Scanner**

Ini adalah salah satu alat yang menonaktifkan label yang paling banyak digunakan. Scanner ini sendiri disesuaikan dengan jenis tag yang dipergunakan. Scanner untuk membaca *barcode* dengan RFID tentu saja berbeda. Namun, fungsi keduanya tetaplah sama.

Ada dua macam bentuk scanner yang digunakan saat ini. Yang pertama bernama flat scanner, yaitu scanner yang dipasangkan pada suatu tempat dan tidak dapat dengan bebas dipindah-pindahkan. Sedangkan satu lagi yang bernama *hand-held scanner* dapat dengan mudah dipindahkan dari satu tempat ke tempat lain. Hand-held scanner lebih mudah menjangkau label di tempat yang lebih sulit, sehingga lebih bekerjanya dapat lebih efektif.

- **Proximity Deactivators (menonaktifkannya secara manual)**

Namun, ada juga tag yang diletakkan secara tersembunyi sehingga untuk menonaktifkannya seseorang perlu untuk melakukannya secara manual. Yaitu dengan memasukkan kodenya pada mesin kasir, lalu dari sini status produk tersebut pun akan berubah yaitu menjadi layak melewati gerbang alarm.

- **Detacher/Releaser**

Alat ini sudah sempat disinggung pada bagian di atas tadi. Ini adalah alat yang digunakan untuk melepaskan tag-tag khusus yang memang tidak mungkin dilepaskan sendiri oleh kustomer. ■

Menangani Masalah TCP/IP pada Windows XP

Gunung Sarjono

(Bagian 1 dari 2 Artikel)

TCP/IP merupakan protokol jaringan yang sangat populer. Walaupun begitu, penggunaannya bukan berarti tanpa masalah. Oleh karena itu, Anda perlu mempersiapkan diri dengan *tool-tool* (program) yang diperlukan.

Bersua lagi di artikel Windows XP. Kali ini kita akan melanjutkan pembahasan protokol TCP/IP pada Windows XP, khususnya dalam menangani masalah protokol jaringan pada Windows XP. Windows XP menawarkan beberapa program miliknya untuk digunakan dalam menangani masalah TCP/IP.

PING

Ping dapat digunakan untuk mengetes koneksi TCP/IP dengan mengirim pesan dari sistem lokal ke *node remote* atau *gateway* (program ini juga dapat digunakan untuk mengetes *loopback* secara lokal untuk melihat apakah ia bekerja dengan baik). Jika node *remote* atau *gateway* menerima pesan tersebut, mereka akan mengirimkan pesan balasan yang terdiri dari alamat IP remote, banyaknya bit, dan waktu yang dibutuhkan untuk merespon (dalam milidetik - ms).

ping [**-t**] [**-a**] [**-n** *Count*] [**-l** *Size*] [**-f**] [**-i** *TTL*] [**-v** *TOS*] [**-r** *Count*] [**-s** *Count*] [{**-j** *HostList* | **-k** *HostList*}] [**-w** *Timeout*] [*TargetName*]

Parameter:

- **-t** Ping host yang dimaksud sampai dihentikan. Untuk melihat statistik dan melanjutkan - gunakan Ctrl-Break; Untuk berhenti - gunakan Ctrl-C.
- **-a** *Resolve*. Alamat ke nama komputer.
- **-n** *Count*. Kirim paket ECHO sebanyak *count*. Default-nya adalah 4.
- **-l** *Size*. Kirim paket ECHO sebesar *size*. Default-nya adalah 32 byte, dan maksimum 65.527 byte.
- **-f** Flag *Don't Fragment*. pada header IP pesan Echo Request yang dikirim diset ke 1. Dengan demikian, pesan Echo Request tidak dapat difragmentasi oleh router yang terdapat pada rute ke tujuan. Parameter ini berguna untuk menangani masalah *Path Maximum Transmission Unit* (PMTU).
- **-i** *TTL*. Menyatakan nilai field TTL pada IP header Pesan Echo Request yang dikirim. Default-nya adalah nilai TTL default untuk host tersebut. Untuk host Windows XP, biasanya 128, dan maksimum 255.
- **-v** *TOS*. Menyatakan nilai field *Type of Service* (TOS) pada IP header pesan Echo Request yang dikirim. Default-nya adalah 0. TOS berupa nilai desimal antara 0 sampai 255.
- **-r** *Count*. Gunakan option *Record Route* pada IP header untuk mencatat path yang diambil oleh pesan Echo Request dan pesan Echo Reply yang sesuai. Setiap hop pada path menggunakan entri pada Record Route, Jika dimungkinkan, gunakan *Count* yang sama atau yang lebih dari jumlah hop antara asal dan tujuan. Count paling sedikit harus bernilai 1 dan maksimum 9.
- **-s** *Count*. Gunakan option *Internet Timestamp* pada IP header untuk mencatat waktu kedatangan pesan Echo Request dan pesan Echo Reply yang sesuai untuk setiap hop. Count paling sedikit harus bernilai 1 dan maksimum 4.
- **-j** *HostList*. Gunakan option *Loose Source Route* pada header IP pesan Echo Request yang kirim ke HostList. Dengan loose source routing, tujuan berturut-turut dapat dipisahkan oleh satu atau beberapa router. Jumlah maksimum alamat atau nama pada host list sebanyak 9. HostList merupakan serangkaian alamat IP (dalam notasi desimal) yang dipisah dengan spasi.
- **-k** *HostList*. Gunakan option *Strict Source Route* pada header IP pesan Echo Request yang kirim ke HostList. Dengan *strict source routing*, tujuan berikutnya harus dapat dihubungi secara langsung (harus *neighbor* atau interface router). Jumlah maksimum alamat atau nama pada HostList adalah 9. HostList merupakan serangkaian alamat IP (dalam notasi desimal) yang dipisah dengan spasi.
- **-w** *Timeout*. Interval *time-out* dalam milidetik. Default-nya adalah 4000 milidetik (4 detik).
- **/?**. Menampilkan help pada command prompt.

ARP

Menampilkan dan mengubah tabel transalasi alamat IP ke alamat fisik Ethernet atau *token ring* yang digunakan oleh *Address Resolution Protocol* (ARP).

arp [**-a** [*InetAddr*] [**-N** *IfaceAddr*]] [**-g** [*InetAddr*] [**-N** *IfaceAddr*]] [**-d** *InetAddr* [*IfaceAddr*]] [**-s** *InetAddr EtherAddr* [*IfaceAddr*]]

Parameter:

- **-a** [*InetAddr*] [**-N** *IfaceAddr*]. Menampilkan tabel cache ARP semua interface. Untuk menampilkan entri ARP alamat IP tertentu, gunakan arp -a dengan parameter *InetAddr*, di mana *InetAddr* adalah alamat IP. Untuk menampilkan entri ARP interface tertentu, gunakan parameter -N *IfaceAddr* dimana *IfaceAddr* merupakan alamat IP interface. Parameter -N merupakan *case-sensitive*.
- **-g** [*InetAddr*] [**-N** *IfaceAddr*]. Identik dengan -a.
- **-d** *InetAddr* [*IfaceAddr*]. Menghapus entri alamat IP tertentu, di mana *InetAddr* adalah alamat IP yang dimaksud. Untuk menghapus entri interface tertentu, gunakan parameter IfaceAddr di mana IfaceAddr adalah alamat IP interface. Untuk menghapus semua entri, gunakan tanda bintang (*) sebagai *InetAddr*.

- **-s** *InetAddr EtherAddr* [*IfaceAddr*]. Menambahkan entri ke cache ARP yang me-*resolve* alamat IP *InetAddr* ke alamat fisik *EtherAddr*. Untuk menambahkan entri cache ARP statis interface tertentu, gunakan parameter *IfaceAddr* di mana *IfaceAddr* adalah alamat IP *interface*.

IPCONFIG

Perintah ini digunakan untuk melihat konfigurasi TCP/IP lokal, seperti alamat IP, subnet mask, dan *gateway default*. *Utility* berguna terutama pada sistem yang menjalankan DHCP, yang memungkinkan *user* untuk melihat konfigurasi yang diberikan oleh DHCP.

ipconfig [**/all**] [**/renew** [*Adapter*]] [**/release** [*Adapter*]] [**/flushdns**] [**/displaydns**] [**/registerdns**] [**/showclassid** *Adapter*] [**/setclassid** *Adapter* [*ClassID*]]

Parameter:

- **/all**. Menampilkan konfigurasi lengkap. Tanpa switch ini, ipconfig hanya akan menampilkan alamat IP, subnet mask, dan gateway default untuk setiap interface jaringan.
- **/renew** [*Adapter*]. Memperbarui parameter konfigurasi DHCP. Option ini hanya tersedia pada sistem yang menjalankan service Client DHCP. Untuk menentukan nama adapter, ketik nama adapter yang tampil ketika Anda menggunakan ipconfig tanpa parameter.
- **/release** [*Adapter*]. Melepas konfigurasi DHCP yang digunakan. Option ini men-*disable* TCP/IP pada sistem lokal dan hanya tersedia pada klien DHCP. Untuk menentukan nama adapter, ketik nama adapter yang tampil ketika Anda menggunakan ipconfig tanpa parameter.
- **/flushdns**. Membersihkan dan me-*reset* isi DNS *client resolver cache*. Selama menangani masalah DNS, Anda dapat menggunakan parameter ini untuk membuang entri negatif dari *cache*, termasuk entri lainnya yang ditambahkan secara dinamis.
- **/displaydns**. Menampilkan isi DNS *client resolver cache*, yang meliputi file Host lokal dan catatan query nama yang baru saja didapat oleh komputer. Service DNS Client menggunakan informasi tersebut untuk me-resolve *query* nama dengan cepat, sebelum menanyakannya ke DNS server.
- **/registerdns**. Melakukan pendaftaran nama DNS dan alamat IP komputer secara manual. Anda dapat menggunakan parameter ini untuk menangani kegagalan pendaftaran nama DNS atau masalah update resolve antara klien dan DNS server tanpa harus me-*reboot* komputer client. Anda dapat melihat nama yang terdaftar di bagian DNS pada *advanced settings* protokol TCP/IP.
- **/showclassid** *Adapter*. Menampilkan DHCP class ID adapter tertentu. Untuk melihat DHCP class ID semua adapter, gunakan tanda bintang (*) pada *Adapter*. Parameter ini hanya tersedia pada komputer yang mendapatkan alamat IP secara otomatis.
- **/setclassid** *Adapter* [*ClassID*]. Mengonfigurasi DHCP class ID adapter tertentu. Untuk mengeset DHCP class ID semua adapter, gunakan tanda bintang (*) pada *Adapter*. Parameter ini hanya tersedia pada komputer yang mendapatkan alamat IP secara otomatis. Jika DHCP class ID tidak ditentukan, maka class ID yang sekarang akan dihilangkan.
- **/?**. Menampilkan help pada *command prompt*.

NBTSTAT

NetBT Statictics digunakan untuk menangani masalah resolusi penamaan NetBIOS over TCP/IP (NetBT). Perintah ini menampilkan statistik protokol dan koneksi TCP/IP yang menggunakan NetBT. NetBT me-*resolve* nama NetBIOS ke alamat IP. Ia menggunakan beberapa option untuk resolusi nama NetBIOS, termasuk pencarian cache lokal, query WINS server, broadcast, pencarian file Lmhosts dan Hosts, dan query DNS server.

nbtstat [**-a** *RemoteName*] [**-A** *IPAddress*] [**-c**] [**-n**] [**-r**] [**-R**] [**-RR**] [**-s**] [**-S**] [*Interval*]

Parameter:

- **-a** *RemoteName*. Menampilkan tabel nama komputer remote dengan menggunakan namanya.
- **-A** *IPAddress*. Menampilkan tabel nama komputer remote dengan menggunakan alamat IP-nya.
- **-c**. Menampilkan tabel nama *cache* NetBIOS remote.
- **-n**. Menampilkan nama NetBIOS lokal. *Registered* berarti nama tersebut terdaftar oleh broadcast (Bnode) atau WINS (jenis node lainnya).
- **-r**. Menampilkan statistik resolusi nama jaringan Windows. Pada komputer Windows XP parameter ini menampilkan jumlah nama yang telah di-*resolve* dan didaftar menggunakan *broadcast* dan WINS.
- **-R**. Membersihkan dan isi cache nama NetBIOS dan kemudian memuat kembali entri #PRE-tag dari file Lmhosts.
- **-RR**. Melepas dan kemudian memperbarui nama NetBIOS komputer lokal yang terdaftar pada WINS server.
- **-s**. Menampilkan sesi client dan server. Alamat IP komputer remote diubah ke namanya dengan menggunakan file Hosts.—**S**. Menampilkan sesi client dan server. Komputer remote hanya ditampilkan dalam alamat IP.
- ***Interval***. Menampilkan kembali konfigurasi setiap interval tertentu. Gunakan Ctrl-C untuk berhenti. Jika tidak digunakan, nbtstat hanya akan menampilkan konfigurasi sebanyak satu kali.
- **/?**. Menampilkan help pada *command prompt*.

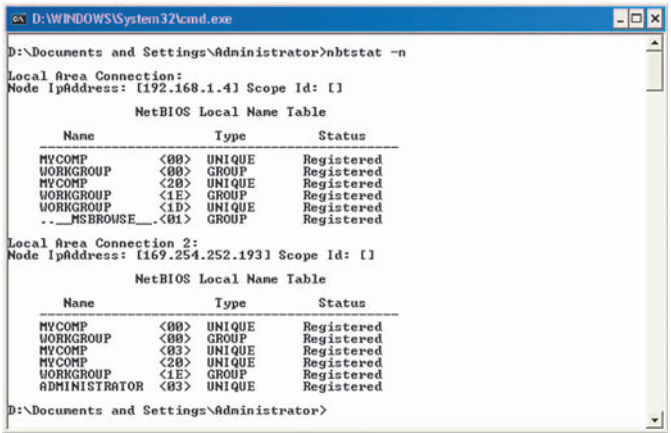
NETSTAT

Netstat menampilkan statistik protokol TCP/IP dan koneksi aktif ke dan dari komputer Anda dan juga option untuk menampilkan jumlah byte yang dikirim dan diterima, termasuk paket jaringan yang *drop* (jika ada).

netstat [**-a**] [**-e**] [**-n**] [**-o**] [**-p** *Protocol*] [**-r**] [**-s**] [*Interval*]

Parameter:

- **-a**. Menampilkan semua koneksi dan



Contoh *output* perintah nbtstat –n.

Contoh *output* perintah pathping.

port yang didengarkan. Koneksi server biasanya tidak ditampilkan.

- **-e**. Menampilkan statistik Ethernet. Option ini dapat dikombinasikan dengan -s.
- **-n**. Menampilkan alamat dan nomor port dalam bentuk numerik (daripada pencarian nama).
- **-o**. Menampilkan koneksi TCP yang aktif, termasuk ID proses (PID) setiap koneksi. Anda dapat menemukan aplikas berdasarkan PID di tab *Process* pada Windows Task Manager. Parameter ini dapat dikombinasikan dengan -a, -n, dan -p.
- **-p** *Protocol*. Menampilkan koneksi protokol pada *procotol* (yang dapat TCP atau UDP). Jika digunakan bersama option -s untuk menampilkan statistik per protokol, maka *protocol* dapat TCP, UDP, ICMP, atau IP.
- **-s**. Menampilkan statistik per protokol. Secara default, statistik protokol TCP, UDP, ICMP, dan IP ditampilkan. Jika protokol IPv6 diinstalasi, maka statistik yang ditampilkan adalah untuk protokol TCP over IPv6, UDP over IPv6, ICMPv6, dan IPv6. Parameter -p dapat digunakan untuk menentukan protokol yang diinginkan.
- **-r**. Menampilkan isi tabel routing. Parameter ini ekivalen dengan perintah *route print*.
- ***Interval***. Menampilkan kembali informasi yang dipilih setiap interval detik tertentu. Gunakan Ctrl-C untuk berhenti. Jika parameter ini tidak digunakan, netstat hanya akan menampilkan konfigurasi tersebut satu kali.
- **/?**. Menampilkan help pada *command prompt*.

ROUTE

Anda dapat menggunakan perintah route untuk menampilkan tabel routing TCP/IP yang digunakan dan menambahkan atau menghapus rute IP.

`route` [**`-f`**] [**`-p`**] [*`Command`* [*`Destination`*] [**`mask`** *`Netmask`*] [*`Gateway`*] [**`metric`** *`Metric`*]] [**`if`** *`Interface`*]]

Parameter:

- **-f**. Membersihkan semua entri tabel routing dari rute yang tidak *host route* (rute dengan netmask 255.255.255.255), rute loopback (rute dengan tujuan 127.0.0.0 dan netmask 255.0.0.0), atau rute *multicast* (rute dengan tujuan 224.0.0.0 dan netmask 240.0.0.0). Jika ini digunakan bersama salah satu *command*, tabel tersebut akan dibersihkan dulu sebelum menjalankan perintah.
- **-p**. Ketika digunakan bersama perintah add, maka akan dibuat rute yang *persistent* antarsistem. Secara default, rute tidak pertahankan ketika sistem di-*restart*. Ketika digunakan bersama perintah print, maka rute persistent yang terdaftar akan ditampilkan.
- *Command*. Salah satu dari *print* (menampilkan rute), *add* (menambahkan rute), *delete* (menghapus rute), dan *change* (mengubah rute yang ada).
- *Destination*. Komputer tujuan.
- **mask** *netmask*. Menentukan subnet mask yang diasosiasikan dengan entri rute. Jika tidak ditentukan, maka subnet mask yang digunakan adalah 255.255. 255.255.
- *Gateway*. Menentukan gateway. Semua nama simbolis yang digunakan untuk *destination* atau *gateway* keduanya mengacu ke file database jaringan bernama Networks, dan file database nama komputer bernama Hosts. Jika perintahnya adalah print atau delete, maka dapat digunakan *wildcard* untuk destination dan gateway, atau gateway bisa tidak digunakan. Jika destination mengandung * atau ?, ia diperlakukan sebagai pola dan hanya rute tujuan yang sesuai yang ditampilkan. Tanda * mewakili string apa pun, dan ? mewakili satu karakter. Sebagai contoh: 157.*.1, *224*.
- **metric** *Metric*. Memberi *cost metric* (dari 1 sampai 9999) yang digunakan untuk menghitung rute yang tercepat, paling andal, dan/atau yang tidak begitu mahal.
- **If** *Interface*. Index interface di mana tujuan dapat dicapai. Untuk semua daftar interface dan indeks yang sesuai, gunakan perintah route print. Anda dapat menggunakan nilai desimal atau heksadesimal. Untuk nilai heksadesimal, didahului dengan 0x. Ketika parameter ini tidak digunakan, interface akan ditentukan dari alamat gateway.
- **/?**. Menampilkan help pada *command prompt*.

HOSTNAME

Hostname digunakan untuk menampil-

kan nama host komputer lokal untuk otentikasi oleh tool *Remote Copy Protocol* (RCP), *Remote Shell* (RSH), dan *Remote Execution* (REXEC).

hostname

Parameter:

/? Menampilkan help pada command prompt.

TRACERT

Tracert kadang-kadang digunakan untuk memeriksa apakah alamat IP klien sudah terkonfigurasi dengan benar. Program ini pada dasarnya akan menampilkan rute yang diambil untuk mencapai sistem remote. Tracert menentukan *path* yang diambil ke tujuan dengan mengirimkan paket *echo Internet Control Message Protocol* (ICMP) ke tujuan dengan *Time-to-Live* (TTL) yang bervariasi. Setiap router sepanjang jalur diperlukan guna menurunkan TTL paket paling sedikit 1 sebelum meneruskannya, jadi TTL tersebut menjadi hitungan hop. Ketika TTL paket mencapai 0, router mengirimkan kembali pesan ICMP Time Exceeded kepada sistem asal.

Tracert menentukan *path* dengan mengirim paket echo pertama dengan TTL 1 dan meningkatkan TTL sebesar 1 pada setiap transmisi berikutnya sampai target memberi respon atau TTL maksimum tercapai. Rute tersebut ditentukan dengan memeriksa pesan ICMP Time Exceeded yang dikirim oleh intermediate router. Namun, beberapa router secara men-drop paket dengan TTL yang kadaluarsa dan ini tidak tampak oleh tracert.

tracert [**-d**] [**-h** *MaximumHops*] [**-j** *HostList*] [**-w** *Timeout*] [*TargetName*]

Parameter:

- **-d**. Mencegah tracert untuk meresolve alamat IP router ke namanya. Hal ini dapat mempercepat hasil tracert.
- **-h** *MaximumHops*. Jumlah hop untuk mencari target (tujuan). Default-nya 30 hop.
- **-j** *HostList*. Gunakan option *Loose Source Route* pada header IP paket Echo Request yang kirim ke HostList. Dengan *loose source routing*, tujuan berturut-turut dapat dipisahkan oleh satu atau beberapa router. Jumlah maksimum alamat atau nama pada HostList adalah 9. HostList merupakan serangkaian alamat IP (dalam notasi desimal) yang dipisah dengan spasi.
- **-w** *Timeout*. Jumlah waktu tunggu (dalam milidetik) ICMP Time Exceeded atau pesan Echo Reply yang akan diterima. Jika tidak diterima setelah *time-out*, maka akan ditampilkan tanda bintang (*). Time-out default selama 4000 milidetik (4 detik).
- *TargetName*. Tujuan yang sesuai dengan alamat IP atau nama host.
- **-?**. Menampilkan help pada *command prompt*.

PATHPING

Pathping juga menampilkan rute yang diambil untuk mencapai sistem remote seperti TRACERT, hanya saja PATHPING melakukannya secara lebih mendetail dan menawarkan fungsionalitas yang lebih banyak. Pathping menyediakan informasi tentang *network latency* dan *network loss* pada hop antara asal dan tujuan. Pathping mengirim beberapa pesan *Echo Request* ke setiap route antara sumber dan tujuan selama periode waktu tertentu dan kemudian menghitung hasilnya berdasarkan paket yang dikembalikan dari setiap router. Karena pathping menampilkan tingkat *packet loss* pada setiap router atau link, Anda dapat mengetahui router atau subnet mana yang jaringannya bermasalah.

Pathping mengidentifikasi router mana yang berada pada rute, dan kemudian mengirimkan ping secara periodik ke semua router selama periode waktu tertentu dan menghitung statistiknya berdasarkan jumlah yang dikembalikan dari setiap router. Jika digunakan tanpa parameter, pathping akan menampilkan *help*.

pathping [**-n**] [**-h** *MaximumHops*] [**-g** *HostList*] [**-p** *Period*] [**-q** *NumQueries* [**-w** *Timeout*] [**-T**] [**-R**] [*TargetName*]

Parameter:

- **-n**. Mencegah pathping untuk tidak me*resolve* alamat IP ke nama host. Hal ini dapat mempercepat hasil pathping.
- **-h** *MaximumHops*. Jumlah hop maksimum untuk mencari target (tujuan). Default-nya 30 hop.
- **-g** *HostList*. Gunakan option *Loose Source Route* pada header IP paket Echo Request yang kirim ke HostList. Dengan *loose source routing*, tujuan berturut-turut dapat dipisahkan oleh satu atau beberapa router. Jumlah maksimum alamat atau nama pada HostList sebanyak 9. HostList merupakan serangkaian alamat IP (dalam notasi desimal) yang dipisah dengan spasi.
- **-p** *Period*. Waktu tunggu antar-ping. Default-nya adalah 250 milidetik (1/4 detik).
- **-q** N*umQueries*. Banyaknya pesan Echo Request yang dikirim ke setiapa router pada path. Default-nya 100 query.
- **-w** *Timeout*. Waktu tunggu untuk setiap balasan. Default-nya 3000 milidetik (3 detik).
- **-T**. Menyertakan tag prioritas layer-2 (misalnya 801.2p) pada pesan Echo Request yang dikirim ke setiap perangkat jaringan pada rute. Hal ini membantu dalam mengidentifikasi perangkat jaringan yang tidak memiliki kemampuan prioritas layer-2. Switch ini digunakan untuk mengetes kualitas koneksi *Quality of Service* (QoS).
- **-R**. Memutuskan apakah setiap perangkat jaringan pada rute mendukung *Resource Reservation Protocol* (RSVP), yang memungkinkan komputer host mencadangkan sejumlah *bandwidth* untuk aliran data. Switch ini digunakan untuk mengetes koneksi *Quality of Service* (QoS).
- *TargetName*. Tujuan yang diidentifikasikan oleh alamat IP atau nama host.
- **/?**. Menampilkan help pada *command prompt*.

Sekian dulu untuk kali ini. Simak artikel berikutnya pada bulan depan untuk mengikuti bagian selanjutnya dan sementara itu, sukses selalu dalam kegiatan Anda. ■